Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tadej Jagodnik

Brezkontaktni sistem plačevanja z mobilnimi NFC napravami

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: doc. dr. Mira Trebar

Ljubljana, 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Tadej Jagodnik, z vpisno številko **63070309**, sem avtor magistrskega dela z naslovom:

Brezkontaktni sistem plačevanja z mobilnimi NFC napravami

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika magistrskega dela, naslov (slovenski, angleški), povzetek (slovenski, angleški) ter ključne besede (slovenske, angleške) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 26. junija 2015 Podpis avtorja:

Zahvaljujem se mentorici doc. dr. Miri Trebar za potrpežljivost, strokovno pomoč, usmerjanje ter nasvete pri izdelavi magistrskega dela. Zahvalil bi se tudi staršem, sestri Tjaši in dekletu Ani za podporo skozi celoten študij.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application Programming Interface	Programski vmesnik
DoS	Denial-of-Service	Napad s preobremenitvijo
HC	Host Controller	Krmilnik gostitelja
HCI	Host Controller Interface	Vmesnik med krmilnikom NFC in HC
HF	High Frequency	Visoka frekvenca
LF	Low Frequency	Nizka frekvenca
LLCP	Logical Link Control Protocol	Protokol za izvedbo P2P
MITM	Man-In-The-Middle	Napad s posrednikom.
NDEF	NFC Data Exchange Format	Format podatkov v NFC
NFC	Near Field Communication	Radiofrekvenčna tehnologija kratkega dosega
NFCIP-1	NFC Interface and Protocol 1	Protokol na fizični plasti tehnologije NFC
NFCIP-2	NFC Interface and Protocol 2	Nadgradnja protokola NFCIP-1
NFC CLF	NFC Contactless Front-end	Brezkontaktni čelni del NFC
NFC-WI	NFC Wired Interface	Vmesnik med krmilnikom NFC in SE

kratica	angleško	slovensko
NPP	NDEF Push Protocol	Protokol za prenos sporočil NDEF
P2P	Peer-to-Peer	Komunikacija vsak z vsakim
PIN	Personal Identification Number	Osebna identifikacijska številka
PoS	Point-of-Sale	Prodajni terminal
RFID	Radio Frequency IDentification	Radiofrekvenčna tehnologija
RSA	Rivest Shamir Adleman	Algoritem za šifriranje z javnim ključem
RTD	Record Type Definition	Tip zapisa sporočila NDEF
SE	Secure Element	Varni element
SMC	Secure Memory Card	Varna pametna in pomnilniška kartica
SNEP	Simple NDEF Exchange Protocol	Protokol za prenos sporočil NDEF
SWP	Single Wire Protocol	Vmesnik med krmilnikom NFC in SE
SWOT	Strengths, Weaknesses, Opportunities and Threats	Analiza prednosti, slabosti, priložnosti in nevarnosti
UDP	User Datagram Protocol	Brezpovezavni protokol za prenos podatkov
UHF	Ultra High Frequency	Ultra visoka frekvenca
UICC	Universal Integrated Circuit Card	Fizična pametna kartica
TCP	Transmission Control Protocol	Povezavno-orientirani protokol za prenos podatkov
TNF	Type Name Format	Struktura podatkovnih polj
XOR	Exclusive Or	Operacija ekskluzivni ali

Povzetek

Mobilne storitve predstavljajo enega od načinov zamenjave klasične plačilne kartice s pametnim telefonom in tehnologijo NFC (Near Field Communication). Pojavljajo se različne ideje kako izvesti gotovinske transakcije med dvema NFC napravama, brez vsakokratne povezave z zalednim sistemom, v katerem ima uporabnik svoj bančni račun. Cilj magistrskega dela je bil razviti sistem za demonstracijo in izvedbo brezkontaktnega plačevanja, ki temelji na uporabi elektronske gotovine. Njegova izvedba je razdeljena na strežniški del, ki predstavlja bančni sistem in uporabniški del, ki omogoča izmenjavo gotovine v nepovezanem načinu. Anonimno plačilo poteka med dvema mobilnima napravama Android s pomočjo komunikacije P2P (Peer-to-Peer) v tehnologiji NFC. Zasnovani in implementirani so postopki, ki omogočajo dvig in polog elektronske gotovine med mobilno aplikacijo in bančnim sistemom. Zagon aplikacije je omogočen z vpisom poznane kode in preverjanjem kartice NFC, varnost podatkov na napravi pa je zagotovljena s šifriranjem. Za detekcijo dvojne porabe je izveden postopek, ki omogoča beleženje podatkov o položeni elektronski gotovini.

Ključne besede

brezkontaktno plačevanje, elektronska gotovina, NFC, P2P, pametne mobilne naprave

Abstract

Mobile services represent one of the possibilities for replacing traditional payment cards with smartphones and NFC (Near Field Communication) technology. There are different ideas how to implement cash transactions between two NFC devices without respective connection with system of a user's bank. The aim of the master's thesis was to develop a system for demonstration and realization of contactless payment, based on electronic cash. The implementation of a system containing the server part which represents banking system, and the user application that enables the exchange of the electronic cash in offline mode. Anonymous payment is performed between two Android mobile devices via P2P (Peer-to-Peer) communication in NFC technology. The procedures for withdrawing and depositing of the electronic cash between the mobile application and the banking system are designed and implemented. It is possible to launch the application by entering the user's PIN code and checking of the NFC card. Data security on a mobile device in ensured with the encryption. The procedure that enables the recording of a data about the deposited electronic cash is carried out and tested with the history data to detect double spending.

Keywords

contactless payment, electronic cash, NFC, P2P, smartphones

Poglavje 1

Uvod

Potreba po hitrejšem, cenejšem in bolj priročnem načinu plačevanja je razlog za nenehen razvoj najrazličnejših oblik plačevanja. S širokim pokritjem računalniških omrežij, dostopnostjo do interneta in razvojem komunikacijskih tehnologij, so se pojavili tudi različni sistemi za elektronsko plačevanje (angl. electronic payment system). Uporaba prvih tovrstnih sistemov sega že desetletja nazaj, vendar so se množično začeli uporabljati šele v zadnjih letih, predvsem zaradi pojava spletnega nakupovanja, možnosti najema računalniških virov v oblakih in hitrejšega ter zanesljivejšega prenosa digitalnih podatkov.

Ljudje so sprejeli uporabo sistemov za elektronsko plačevanje zaradi hitrosti in enostavnejše uporabe, kar se odraža pri zmanjševanju števila gotovinskih (angl. cash) plačil. Gotovina je nepriročna pri prenašanju in shranjevanju, predvsem v velikih količinah. Plačevanje z njo je zamudno, saj jo je potrebno pred izvedbo plačila pridobiti v banki ali bančnem avtomatu. Elektronsko plačevanje to poenostavi, saj je v večini primerov potrebno prenašati le pametno bančno kartico (plačilna, kreditna, debetna itd.).

Razvoj sistemov za elektronsko plačevanje je dodatno pospešil pojav pametnih mobilnih naprav (angl. smartphone). Tovrstne naprave imajo veliko procesorsko moč, pomnilnik in nabor različnih komunikacijskih tehnologij. Zaradi svoje učinkovitosti in priročnosti jih ljudje ves čas nosijo s seboj in jih

lahko povsod uporabljajo. Postale so del sistemov, ki jih uvrščamo v področje vseprisotnega računanja (angl. ubiquitous computing), kamor se uvršča tudi elektronsko plačevanje. Naprave v povezavi s komunikacijo bližnjega dosega (Near Field Communication – NFC) so primerne za uporabo v sistemih za brezkontaktno (angl. contactless) plačevanje v povezavi s terminalom PoS (Point of Sale). Ker omogočajo emulacijo brezkontaktnih plačilnih kartic, še bolj poenostavijo plačevanje, saj odstranijo potrebo po prenašanju in uporabi kartice. Obstaja kar nekaj komercialnih mobilnih sistemov, kot so Google Wallet, Apple Pay in Visa payWave in tudi raziskovalnih kot je IDA-Pay.

Kljub omenjenim pomanjkljivostim gotovine in zmanjševanju njene uporabe, je še vedno prisotna v vsakdanjih transakcijah z manjšimi zneski. Predvsem zato, ker gre za poenostavljen sistem za plačevanje, uporaba katerega se ljudem zdi samoumevna in so nanj navajeni. Razlog je tudi zasebnost (angl. privacy) in anonimnost (angl. anonymity) plačnika in tudi prejemnika plačila, ki ju sistemi za elektronsko plačevanje ne zagotavljajo. Poleg tega nekateri uporabniki (predvsem trgovci) še vedno ne uporabljajo sistemov za elektronsko plačevanje (nimajo terminalov PoS), ker banke zaračunajo stroške njihove uporabe. Dodatno obstoječi sistemi za elektronsko plačevanje tipično uporabniku omogočajo le izvedbo plačila, ne omogočajo pa prejema plačila, kar gotovina omogoča.

Namen magistrskega dela je analizirati obstoječe plačilne sisteme, med njimi tudi mobilne, kjer so naprave NFC uporabljene kot emulator kreditne kartice pri izvedbi transakcije na plačilnem terminalu PoS. Razložiti varne načine delovanja ob upoštevanju zasebnosti, opisati komunikacijske protokole za izmenjavo sporočil pri brezkontaktnem prenosu podatkov in podati pričakovane omejitve, napade in tveganja v primeru komunikacije kratkega dosega. Cilj magistrskega dela je izvedba brezkontaktnega sistema plačevanja med dvema napravama NFC v načinu vsak z vsakim (Peer to Peer – P2P), ki je zasnovan in realiziran v povezavi z elektronsko gotovino (angl. electronic cash, e-cash). Določiti je potrebno zasnovo sistema in njegovo arhitekturo ter definirati postopek, ki omogoča transakcijo elektronske gotovine iz ene na-

prave na drugo v nepovezanem načinu (angl. offline). Glavni del je mobilna aplikacija za naprave NFC z operacijskim sistemom Android. Poleg tega je bila razvita strežniška aplikacija, kar omogoča izvedbo celovitega sistema za analizo in empirično vrednotenje rešitve z uporabo analize SWOT (Strengths, Weaknesses, Opportunities and Threats). Opredeljene ugotovitve bodo uporabljene za nadaljnje raziskave na področju elektronskega plačevanja in tehnologije NFC.

V Poglavju 2 je podan pregled področja, sestavljen iz opisa plačilnih sistemov in tehnologije NFC. V prvem delu so pojasnjeni osnovni koncepti plačevanja in plačilnih sistemov v splošnem. Predstavljeni so udeleženci, ki v njih sodelujejo in njihove naloge. Opisani so sistemi za elektronsko plačevanje, kjer je pojasnjena razlika med povezanim (angl. online) in nepovezanim (angl. offline) plačevanjem. Poudarek je predvsem na elektronski gotovini, njenih lastnostih, varnostnih zahtevah in mehanizmih s katerimi jo je mogoče realizirati. Drugi del pregleda področja predstavlja tehnologija NFC in arhitektura mobilnih naprav Android. Predstavljen je osnovni koncept tehnologije NFC, njeni protokoli in standardi ter njena uporaba na platformi Android. V Poglavju 3 je predstavljena predlagana zasnova in arhitektura sistema za brezkontaktno elektronsko plačevanje. Opisana je inicializacija sistema, postopek za ustvarjanje e-računa (eR), dvig, plačilo in polog elektronske gotovine. V Poglavjih 4 in 5 je predstavljen potek implementacije, opis testiranja sistema in njegove analize v tabeli SWOT. V zaključku je na kratko povzeto opravljeno delo, poudarjene so ugotovitve in podane smernice za nadaljnje delo ter možne izboljšave.

Poglavje 2

Pregled področja

Osnovni koncepti sistemov za plačevanje in tehnologija NFC predstavljajo osnove za razvoj sistema za brezkontaktno plačevanje. Prvi del sestavljajo opisi različnih oblik plačevanja in predstavitev pomena elektronskega plačevanja, kjer je pojasnjena razlika med povezanim in nepovezanim plačevanjem. Poudarek je predvsem na elektronski gotovini, kjer so opisane njene lastnosti, varnostne zahteve in možne zlorabe. Podani so kriptografski mehanizmi, ki omogočajo njeno izvedbo, doseganje želenih lastnosti in preprečevanje zlorab. Podana je osnovna shema za plačevanje z elektronsko gotovino, sestavljena iz postopkov za dvig, plačilo in polog. V drugem delu je predstavljena tehnologija NFC in arhitektura mobilnih naprav NFC. Opisana je delitev naprav in komunikacije NFC ter trije možni načini delovanja. Predstavljena so standardizacijska telesa tehnologije in njihove rešitve. Poudarek je na komunikaciji P2P, njeni izvedbi, uporabljenem formatu podatkov in varnosti. Na koncu je predstavljena platforma Android v povezavi s tehnologijo NFC.

2.1 Plačilni sistemi

Plačevanje (angl. payment) obstaja že od nastanka prvih civilizacij in se uporablja v zameno za različno blago (angl. goods) ali opravljeno storitev (angl. service) [14, 1]. Prvotna oblika je bila neposredna zamenjava blaga ali

storitve za neko drugo blago ali storitev. Sledil je pojav prve oblike denarja (angl. money) v blagovni obliki. Za izvedbo plačil so se uporabljale različne surovine (zlato, srebro, sol itd.), katerih vrednost je bila vsem dobro znana.

Zaradi neučinkovitosti in nepriročnosti se je pojavila potreba po večji abstraktni predstavitvi vrednosti in posledično uvedbi gotovine (angl. cash). Uporabljajo se bankovci (angl. note) in kovanci (angl. coin) različnih vrednosti. Tovrstni način je zaradi učinkovitosti in enostavnosti postal najbolj priljubljena oblika plačevanja za manjše količine gotovine. Glavne lastnosti gotovine so:

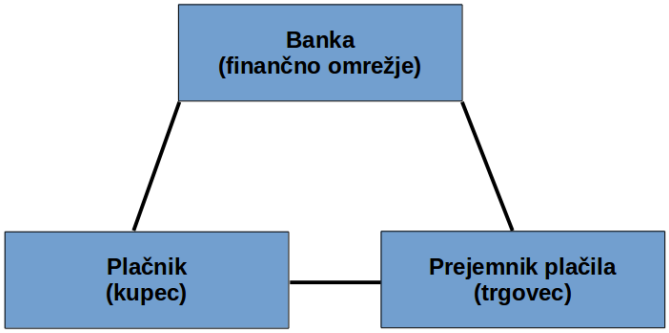
- **Sprejemljivost (angl. acceptability)** – gotovina je univerzalna oblika plačila, ne glede na znesek.
- **Zagotovljeno plačilo (angl. guaranteed payment)** – fizična izmenjava gotovine preprečuje kasnejše nespoštovanje plačila.
- **Brez provizij (angl. no fees)** – izmenjava gotovine poteka brez dodatnih stroškov ali provizij.
- **Anonimnost (angl. anonymity)** – plačilo z gotovino je za razliko od drugih oblik plačevanja anonimno in ne pušča sledi.

Pomanjkljivost gotovine je varno hranjenje, predvsem pri večjih količinah. Rešitev predstavljajo finančne ustanove, imenovane tudi banke (angl. bank), kjer uporabniki hranijo svoja finančna sredstva na bančnih računih. Poleg hranjenja zagotavljajo svojim uporabnikom tudi druge finančne storitve, med drugim različne oblike plačevanja. Bančni računi so osnova za ves nadaljnji razvoj plačilnih sistemov. Banka pa je vključena v vseh oblikah plačevanja, tudi v gotovinskem, saj mora uporabnik gotovino pred samim plačilom pridobiti iz svojega transakcijskega računa v banki ali na bančnem avtomatu.

Poenostavljeni sistemi za plačevanje so sestavljeni iz naslednjih treh udeležencev (slika 2.1):

- **plačnik (angl. payer)** ali kupec (angl. buyer),

- **prejemnik plačila (angl. payee)** ali trgovec (angl. merchant) in
- **banka (angl. bank)** ali finančno omrežje (angl. financial network).



Slika 2.1: Udeleženci v plačilnih sistemih.

Izmenjava finančnih sredstev med plačnikom in prejemnikom plačila se imenuje tudi transakcija (angl. transaction). Vlogi plačnika in prejemnika plačila se lahko velikokrat tudi zamenjata. V poenostavljeni obliki sistema za plačevanje sta plačnik in prejemnik plačila uporabnika iste, v praksi pa velikokrat uporabnika različnih bank. V slednjem primeru v sistemu nastopata dve banki, ki poskrbita za izvedbo plačila. Plačilni sistemi so razdeljeni v naslednje tri skupine:

- **Predplačniški sistemi (angl. prepaid)**, kjer so finančna sredstva odvzeta plačniku pred izvedbo nakupa.
- **Sistemi s takojšnjim plačilom (angl. immediate)**, kjer so finančna sredstva plačniku odvzeta med nakupom.
- **Sistemi s kasnejšim plačilom (angl. subsequent)**, kjer so finančna sredstva plačniku odvzeta po opravljenem nakupu.

Poleg gotovinskega plačevanja in plačevanja s pomočjo čekov (angl. cheque) potekajo ostali načini plačevanja v elektronski obliki. Elektronsko plačevanje (angl. electronic payment, e-payment) je del elektronskega poslovanja (angl. electronic commerce, e-commerce) in se nanaša na katerokoli finančno transakcijo, ki zajema prenos podatkov v digitalni obliki.

2.1.1 Povezano in nepovezano plačevanje

Sistemi za plačevanje se glede na udeleženost banke v postopku plačila delijo na [10]:

- **povezane (angl. online)** in
- **nepovezane (angl. offline)**.

Pri povezanih sistemih je banka vedno vključena v postopek plačila med plačnikom in prejemnikom plačila. Na takšen način lahko sproti preverja legitimnost uporabnika in veljavnost plačila. Preverjanje plačila v nepovezanih sistemih je delno izvedeno le s strani prejemnika plačila. Ta lahko preveri znesek, ustreznost formata in obliko plačila. Popolno preverjanje veljavnosti plačila se izvede šele, ko slednji želi unovčiti prejeta finančna sredstva in se poveže z banko. Banka pri izvedbi plačila ne sodeluje, zato povezava s finančnim omrežjem ni potrebna. Povezani sistemi imajo dve večji pomanjkljivosti oziroma slabosti. Prva je odvisnost od tehnologije in komunikacije, saj v primeru težav s povezavo, izvedba plačila ni mogoča. Druga pa neanonimnost plačnika oziroma sledljivost plačilu. Banka ve, kdo je sodeloval v plačilu in hrani zgodovino vseh plačil.

2.1.2 Varnost

Varnost je za sisteme za elektronsko plačevanje ključnega pomena [10]. Zagotovljene morajo biti naslednje splošne varnostne zahteve, ki so skupne vsem vrstam elektronskega plačevanja:

- **Zasebnost (angl. privacy)**, ki predstavlja zaščito pred prisluškovanjem.
- **Avtentikacija (angl. authentication)**, ki je sestavljena iz:
  - identifikacije uporabnikov (angl. user identification), ki predstavlja zaščito pred lažnim predstavljanjem in

  - integritete (angl. integrity) sporočil – zaščita pred spreminjanjem vsebine sporočil.

- **Nezanikanje (angl. nonrepudation)**, ki omogoča zaščito pred kasnejšim zanikanjem plačila.
- **Singularnost (angl. singularity)** plačila, kar pomeni, da ni mogoče uporabiti istega plačila večkrat.
- **Zaupnost (angl. confidentiality)**, kjer so vse informacije na voljo le udeležencem v plačilu.
- **Razpoložljivost (angl. availability) in zanesljivost (angl. reliability)**, ki omogočata vsem udeležencem izvedbo ali prejem plačila v celoti.

2.1.3 Elektronska gotovina

Elektronska gotovina (angl. electronic cash, e-cash) je plačilni instrument v elektronski obliki, ki posnema poslovanje z gotovino [10, 3]. Z njeno uporabo se ohranja anonimnost plačnika in nesledljivost plačila s strani banke. Digitalni zapis s katerim je predstavljena elektronska gotovina, se imenuje žeton (angl. token) ali kovanec. Način njunega zapisa je enak kot pri katerem koli drugem digitalnem podatku.

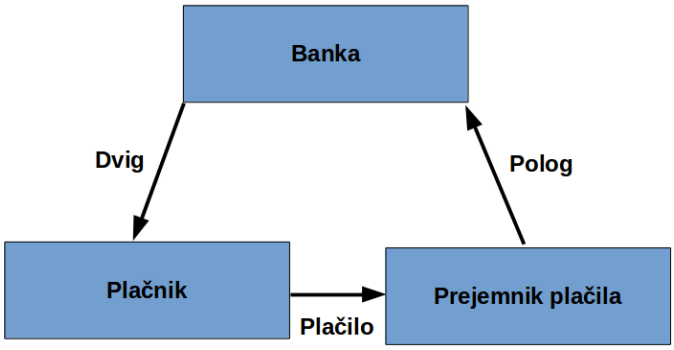
Želja snovalcev in tudi uporabnikov elektronske gotovine je, da bi imela enake lastnosti kot jih ima gotovina, vendar vseh ni mogoče prenesti v digitalni svet. Lastnosti idealne elektronske gotovine so naslednje [13]:

- **Neodvisnost (angl. independence)** – varnost elektronske gotovine je neodvisna od njene lokacije.
- **Varnost (angl. security)** – elektronske gotovine ni mogoče zlorabiti.
- **Anonimnost (angl. anonymity)** – zasebnost (angl. privacy) uporabnikov in **nesledljivost (angl. untraceability)** elektronske gotovine.

- **Prenosljivost (angl. transferability)** – elektronsko gotovino je lahko prenesti iz enega uporabnika na drugega.
- **Deljivost (angl. dividability)** – elektronsko gotovino je mogoče razdeliti na več delov, ki predstavljajo manjše vrednosti.

Udeleženci, ki sodelujejo v plačilni shemi so uporabniki (plačniki in prejemniki plačila) in banka. Uporabniki imajo na banki odprte bančne račune. Uporaba elektronske gotovine je mogoča s pomočjo treh postopkov. Vsak od njih je avtonomen in se izvede med dvema udeležencema (slika 2.2):

- **Dvig (angl. withdraw)** – banka posreduje gotovino udeležencu.
- **Plačilo (angl. payment)** – udeleženca izvedeta prenos gotovine.
- **Polog (angl. deposit)** – udeleženec posreduje gotovino banki.



Slika 2.2: Postopki, ki omogočajo uporabo elektronske gotovine.

Za pridobitev gotovine se uporablja postopek za dvig. Med plačnikom in prejemnikom plačila se izvede postopek za plačilo v povezanem ali nepovezanem načinu. Prejemnik plačila lahko elektronsko gotovino unovči s postopkom za polog. Izvede ga lahko kadarkoli, takoj po zaključenem plačilu ali pa kasneje. Polog se vedno izvrši za celoten znesek elektronske gotovine, ki ga ima udeleženec. Postopka za dvig in polog se izvedeta med udeležencem in banko.

Sistemi za plačevanje z elektronsko obliko, se prav tako kot vse ostale oblike srečujejo s potencialnimi zlorabami ali kriminalnimi dejanji. Pri uporabi elektronske gotovine sta pomembni:

- **Ponarejanje (angl. counterfeiting)** – ustvarjanje navidez veljavne elektronske gotovine brez izvedbe ustreznega postopka za dvig.
- **Večkratna poraba (angl. multiple spending) ali pogostejše dvojna poraba (angl. double spending)** – večkratna uporaba iste elektronske gotovine.

Ponarejanje je mogoče preprečiti z uporabo avtentikacije, ki omogoča identifikacijo uporabnika in integriteto sporočil. Problem dvojne porabe je veliko težji. Gre za vrsto zlorabe, ki je pogosta pri shemah za elektronsko plačevanje, ki temeljijo na uporabi elektronske gotovine [10]. Nanaša se na uporabo ene enote elektronske gotovine več kot enkrat. Z razliko od gotovine je elektronsko enostavno podvojiti (angl. duplicate) s kopiranjem ali z njeno ohranitvijo po opravljenem plačilu. Dvojno porabo je lažje zaznati kot pa preprečiti.
Banka vzdržuje podatkovno bazo že porabljene elektronske gotovine, na podlagi katere lahko določi ali je bila neka enota elektronske gotovine že porabljena ali ne. V povezanih plačilnih sistemih je preprečitev dvojne porabe mogoča, ker lahko banka zavrne elektronsko gotovino med samim plačilom. Sicer je preprečitev dvojne porabe odvisna od varnosti strojne opreme in zahteva uporabo zaupanja vredne strojne opreme (angl. trusted hardware). Stanje takšne naprave ni mogoče spremeniti ali ponarediti (angl. tampered). Brez takšne strojne opreme je enostavno shraniti njeno trenutno stanje, izvesti plačilo, obnoviti prejšnje stanje in ponovno izvesti plačilo. V praksi je izvedba tovrstne strojne opreme, ki bi se lahko upirala resnim napadom skoraj nemogoča. Eno izmed naprav, ki naj bi preprečila spreminjanje in kopiranje je predstavil David Chaum, ki velja za pionirja in očeta elektronskega plačevanja ter elektronske gotovine [3]. Ker mobilne naprave, uporabljene pri našem delu ne omogočajo tovrstne strojne opreme, preprečitev dvojne po-

rabe ni mogoča.
Mogoča je zgolj zaznava dvojne porabe elektronske gotovine in razkritje identitete uporabnika oziroma storilca. Anonimnost uporabnika je preklicana le v primeru storitve dvojne porabe. V primeru enkratne porabe pa ni razkrita nikakršna informacija o sami elektronski gotovini ali o uporabnikovi identiteti.

2.1.3.1 Shema za plačevanje z elektronsko gotovino

Osnovna anonimna shema za plačevanje z elektronsko gotovino je sestavljena iz postopkov za inicializacijo sistema, odprtje računa, dvig, plačilo in polog [5]. Postopek inicializacije bančnega sistema izvede banka le enkrat ob prvi vzpostavitvi sistema. Slednji zajema določitev njenega javnega in zasebnega ključa ter parametrov potrebnih za uporabo sistema. Postopek odprtja računa se izvede med banko in njenim uporabnikom (Alice in Bob), ko se slednji odloči uporabljati storitev za plačevanje z elektronsko gotovino. Bančni sistem mu priskrbi potrebne podatke, ki omogočajo uporabo te storitve. Alice (plačnik) izvede postopek za dvig elektronske gotovine v povezavi z banko, na osnovi njenega transakcijskega računa:

1. Alice ustvari n anonimnih sporočil, kjer vsak izmed njih predstavlja potencialno enoto elektronske gotovine. Zapis je sestavljen iz:
   - vrednosti,
   - naključnega unikatnega niza x, ki je dovolj dolg, da je verjetnost za pojav dveh enakih zanemarljiva in
   - seznama sestavljenega iz n parov identitet, ki so pridobljeni s pomočjo metode za razdelitev skrivnosti (angl. secret splitting).
2. Alice zakrije (angl. blind) vseh n sporočil s pomočjo protokola za izvedbo slepih podpisov (angl. blind signature protocol).
3. Alice posreduje vseh n zakritih sporočil banki.

4 Banka naključno izbere $n-1$ sporočil in zahteva od Alice, da jih razkrije (angl. unblind).

5. Alice razkrije izbranih $n-1$ sporočil in jih posreduje banki.

6. Banka preveri format sporočil, njihove vrednosti, identifikatorje oziroma naključne unikatne nize in prisotnost skrite identitete v vseh elementih seznama.

7. Če je preverjanje uspešno, banka digitalno podpiše preostalo zakrito sporočilo.

8. Banka posreduje digitalno podpisano zakrito sporočilo Alice in bremeni njen transakcijski račun.

10. Alice razkrije prejeto digitalno podpisano sporočilo.

Pomembno je, da se med postopkom za dvig ne razkrije identifikator oziroma naključen unikaten niz x elektronske gotovine, kar zagotavlja anonimnost. Alice dvignjeno gotovino hrani v navidezni denarnici na svoji napravi, s katero lahko kasneje izvede plačilo Bobu (prejemnik plačila):

11. Alice posreduje elektronsko gotovino (digitalno podpisano razkrito sporočilo) Bobu.

12. Bob preveri veljavnost elektronske gotovine s pomočjo javnega ključa banke.

13. Bob pošlje Alice izziv – seznam dolžine n, sestavljen iz naključnih vrednosti 0 ali 1. Posamezna vrednost pove Alice kateri del para identitete mora razkriti. Če je vrednost 0 razkrije levi, če ne pa desni del.

14. Alice posreduje Bobu odgovor na izziv – seznam dolžine n, sestavljen iz n delov (levih ali desnih) parov identitete.

Bob lahko unovči elektronsko gotovino tako, da jo posreduje banki s postopkom za polog:

15. Bob posreduje vso prejeto elektronsko gotovino, izzive in pripadajoče odgovore banki.

16. Banka preveri veljavnost elektronske gotovine s svojim javnim ključem.

17. Banka preveri dvojno porabo s pomočjo identifikatorja oziroma naključnega unikatnega niza x:

18. a.) Če v podatkovni bazi še ne obstaja predhodno položena elektronska gotovina z enakim identifikatorjem, potem je polog uspešen. Banka poveča Bobu stanje transakcijskega računa in shrani elektronsko gotovino skupaj z izzivi in pripadajočimi odgovori v podatkovno bazo.

18. b.) Če v podatkovni bazi že obstaja predhodno položena elektronska gotovina z enakim identifikatorjem potem, je polog neuspešen. Sledi postopek odkrivanja identitete storilca dvojne porabe in preklic njegove anonimnosti.

Ob vsakem pologu banka preveri prisotnost elektronske gotovine v podatkovni bazi na podlagi identifikatorja oziroma naključnega unikatnega niza x. Ko Alice prvič porabi enoto elektronske gotovine je njeno preverjanje v postopku pologa uspešno. Slednja se nato shrani v podatkovno bazo skupaj z izzivom in pripadajočim odgovorom, ki ga naključno ustvari Bob. V primeru ko to stori drugič, je preverjanje neuspešno, saj se elektronska gotovina z istim unikatnim identifikatorjem že nahaja v podatkovni bazi. Tako banka ve, da je prišlo do dvojne porabe. Banka lahko ugotovi identiteto storilca dvojne porabe (Alice) in prekliče njegovo anonimnost na podlagi odgovorov na izziva. Pomembno je, da je Bob v prvem in drugem plačilu ustvaril naključna izziva, ki sta med seboj različna. Ugotavljanje identitete poteka s primerjanjem istoležnih elementov odgovorov, kjer vsak predstavlja del identitete Alice. V primeru, ko najde tako levi kot desni del identitete (tam kjer je bila vrednost izziva enkrat 0, drugič 1), lahko z izvedbo operacije XOR ugotovi identiteto storilca dvoje porabe (Alice). Verjetnost, da Bob v dveh postopkih plačila ustvari različna izziva je enaka $1 - (1/2)^n$.

Pri ustvarjanju elektronske gotovine lahko načeloma pride do pojava dveh enakih identifikatorjev oziroma naključnih unikatnih nizov x. V tem primeru bi banka zaznala dvojno porabo, čeprav do nje ni prišlo. Verjetnost, da pri ustvarjanju identifikatorja pride do pojava dveh enakih je enaka $1/2^n$.
Alice prav tako ne more spremeniti unikatnega identifikatorja ali seznama identitete, saj digitalni podpis banke nad elektronsko gotovino nebi bil več veljaven. Slednje lahko Bob ugotovi že med samim plačilom s preverjanjem digitalnega podpisa z javnim ključem banke.
Tudi Bob ne more položiti elektronske gotovine več kot enkrat, saj banka enako kot zgoraj ugotovi prisotnost identifikatorja v podatkovni bazi. Prav tako ne more položiti elektronske gotovine dvakrat in zato obtožiti Alice, saj le ona lahko sestavi odgovor na izziv, sestavljen iz delov parov identitete.
Med dvigom lahko Alice na skrivaj spremeni vrednost elektronske gotovine ali v njo vključi identiteto nekoga drugega. Verjetnost, da Alice goljufa med dvigom elektronske gotovine je enak $1/n$.
Zgornja verjetnost ni zanemarljiva, predvsem pri velikem obtoku elektronske gotovine. Z večanjem števila n se zmanjša možnost goljufanja, vendar se zmanjša tudi učinkovitost sistema. Dodatno rešitev za zmanjšanje tovrstnega goljufanja med dvigom predstavlja uvedba različnih kazni in sankcij, ki bi doletele morebitnega storilca.

2.1.3.2 Kriptografski mehanizmi

Izvedba varnega sistema za elektronsko plačevanje in uporaba elektronske gotovine sta mogoča z uporabo različnih kriptografskih mehanizmov in metod ter predpostavk [10]. Z njihovo pomočjo je mogoče zagotoviti varnostne zahteve in doseči želene lastnosti sistemov.

Simetrična kriptografija (angl. symmetric cryptography) uporablja za šifriranje in dešifriranje sporočila isti ključ [15]. Ključ v simetrični kriptografiji se imenuje tudi deljena skrivnost (angl. shared secret), katero lahko poznata le udeleženca, ki sodelujeta v komunikaciji.

Asimetrična kriptografija (angl. asymmetric cryptography) ali

kriptografija z javnim ključem (angl. public-key) omogoča avtentikacijo, ki vključuje identifikacijo uporabnika in integriteto sporočil [15]. Šifriranje in dešifriranje sporočila sta ločena s pomočjo uporabe dveh ključev, javnega in zasebnega. Javni ključ je na razpolago komurkoli, zasebni pa le lastniku in mora biti varno shranjen. Za šifriranje sporočila se uporablja javni ključ, dešifriranje sporočila pa je mogoče le z zasebnim ključem.

Uporaba asimetrične kriptografije omogoča izvedbo **digitalnega podpisa (angl. digital signature)**, ki omogoča zaupnost, identifikacijo uporabnika, integriteto in nezanikanje sporočila [15]. Z njegovo pomočjo lahko prejemnik sporočila preveri ali je bilo poslano s strani pravega pošiljatelja. Uporabnik se identificira tako, da dokaže poznavanje svojega zasebnega ključa, brez njegovega razkritja. Tisti, ki je sporočilo podpisal ne more zanikati, da je on tudi pošiljatelj sporočila, saj le on pozna zasebni ključ. Preverjanje digitalnega podpisa je ključnega pomena za zagotavljanje, da sporočilo ni bilo med pošiljanjem spremenjeno (integriteta). Z njim lahko udeleženci sistema za plačevanje preverijo veljavnost in sprejemljivost plačila.

Digitalni podpisi zahtevajo uporabo varnih **zgoščevalnih funkcij (angl. hash function)**, imenovane tudi **kriptografske zgoščevalne funkcije (angl. cryptographic hash function)** [15]. Razlog je v preprečevanju ponarejanja podpisov oziroma izgradnji digitalnega podpisa brez zasebnega ključa in zavračanja podpisa sporočila. Zgoščevalna funkcija preslika katerikoli niz bitov v drug niz bitov fiksne dolžine. Predpostavljeno je, da je varna zgoščevalna funkcija enosmerna.

Enosmerna funkcija (angl. one-way function) zagotavlja integriteto in deluje na predpostavki, da jo je hitro in enostavno izračunati $f(a) = b$, s podano funkcijo f in vhodom a [15]. S podano funkcijo f in vhodom b, pa je zelo težko nazaj izračunati $f^{-}1(b) = a$. Pomembna lastnost enosmerne funkcije je, da nima kolizij (angl. collision). Torej mora biti težko ali nemogoče najti dve vhodni vrednosti, ki na izhodu vrneta enako vrednost.

Postopek standardnega algoritma za kriptografijo z javnim ključem RSA (Rivest Shamir Adleman), ki temelji na faktorizaciji zelo velikih naravnih

števil je naslednji:

1. Izbira dveh velikih med seboj različnih naravnih števil p in q.

2. Izračun modula:
$$n = p * q. \tag{2.1}$$

3. Izračun Eulerjeve funkcije:
$$\phi(n) = (p-1) * (q-1). \tag{2.2}$$

4. Naključna izbira šifrirnega ključa e je določena tako, da je njun največji skupni delitelj enak 1:
$$gcd(e, \phi(n) = 1). \tag{2.3}$$

5. Izračun dešifriranega ključa d:
$$d = e^{-1} \mod \phi(n). \tag{2.4}$$

Šifriranje in dešifriranje sporočila M poteka na naslednji način:

1. Sporočilo M se razdeli na manjše dele, kjer je vsak predstavljen kot celo število. Celo število mora biti manjše kot n, za kar poskrbi velikost bloka.

2. Šifrirano sporočilo C je izračunano kot:
$$C = M^e \mod n. \tag{2.5}$$

3. Dešifrirano sporočilo M je izračunano kot:
$$M = C^d \mod n. \tag{2.6}$$

Slepi podpisi (angl. blind signature) omogočajo izvedbo digitalnega podpisa nad sporočilom, brez razkritja njegove vsebine [15, 4]. Predstavil jih je David Chaum in se uporabljajo za izvedbo anonimnega in nesledljivega plačila. Postopek izvedbe slepega podpisa, ki temelji na RSA je naslednji:

1. Alice pomnoži sporočilo z naključno vrednostjo k (med 1 in n), imenovano skrivni faktor (angl. blinding factor). Rezultat je slepo sporočilo, katerega vsebino ni mogoče prebrati:

$$T = M * k^e \mod n, \tag{2.7}$$

   kjer je M sporočilo in e Bobov javni ključ ter n javni modul.

2. Alice pošlje slepo sporočilo Bobu.

3. Bob digitalno podpiše slepo sporočilo:

$$T^d = (M * k^e) \mod n = M^d * k \mod n, \tag{2.8}$$

   kjer je d Bobov privatni ključ.

4. Bob pošlje digitalno podpisano slepo sporočilo Alice.

5. Alice razkrije slepo podpisano digitalno sporočilo (deli ga s skrivnim faktorjem k):

$$S = T^d/k = M^d * k/k \mod n. \tag{2.9}$$

   Rezultat je prvotno sporočilo, ki je digitalno podpisano s strani Boba:

$$S = M^d \mod n. \tag{2.10}$$

Uporaba slepih digitalnih podpisov mora zagotavljati:

- Skrivnost (angl. blindless) – vsebina sporočila mora biti skrita tistemu, ki jo podpisuje.
- Ponarejanje slepih digitalnih podpisov ni mogoče.
- Izhod končnega digitalnega podpisa ni mogoče povezati z uporabnikom.

Za detekcijo dvojne porabe in za zagotavljanje singularnosti plačila se uporabljata metodi **razreži in izberi (angl. cut-and-choose)** in metoda za **razdelitev skrivnosti (angl. secret splitting)** [15, 10]. Omogočata

vključitev identitete uporabnika v plačilo (elektronsko gotovino) in je namenjena razkritju osebe, ki je izvedla dvojno porabo. V posamezno elektronsko gotovino se vključi med postopkom za dvig. Metodi temeljita na razdelitvi sporočila m na dva dela, kjer vsak posebej o le-tem ne razkrije ničesar, oba skupaj pa data sporočilu celotno sliko. Ustvari se naključni niz r, enake dolžine kot je sporočilo m. Izračun novega niza s je mogoč s pomočjo operacije XOR:

$$s = m \oplus r. \tag{2.11}$$

Par je sestavljen iz naključnega niza r in izračunanega niza s, ki skupaj s pomočjo operacije XOR razkrijeta sporočilo m:

$$m = r \oplus s. \tag{2.12}$$

Seznam identitete, ki ga je med pologom potrebno vključiti v ustvarjena slepa sporočila, mora vsebovati n parov. Vsak par je sestavljen iz naključno ustvarjenega niza r in niza s na enak način kot zgoraj. Tako je vsak par v seznamu različen, vendar razkrije isto informacijo. Med dvigom je potrebno banki razkriti izbranih $n - 1$ sporočil in s tem tudi informacije o identiteti. Banka mora za vseh $n - 1$ preveriti prisotnost in razkrito identiteto (na podlagi nizov r in s) vseh n parov.
Med izvedbo plačila, prejemnik plačila ustvari izziv dolžine n, ki je naključen niz dolžine n ter je sestavljen iz ničel in enic. Če se izziv začne z 011, potem mora plačnik iz prvega para razkriti niz r, iz drugega niz s, iz tretjega prav tako niz s itd. Tako sestavi odgovor dolžine n, ki ga prejemnik plačila v postopku pologa skupaj z izzivom posreduje banki. V primeru dvojne porabe prejemnik plačila ponovno ustvari naključen izziv, ki se od prejšnjega razlikuje v vsaj enem bitu. Ko banka v postopku pologa prejme že porabljeno elektronsko gotovino, lahko na podlagi obeh odgovorov ugotovi identiteto storilca dvojne porabe. Na istoležnih mestih, kjer sta se izziva razlikovala, se razlikujeta tudi odgovora. Tako banka pridobi oba pripadajoča dela r in s ter izračuna informacijo o identiteti.

Za večino zgornjih kriptografskih mehanizmov je pomembna uporaba varnih in močnih **generatorjev naključnih števil (angl. random number**

generator) [15]. Ti se delijo na prave generatorje naključnih števil (angl. true random number generator), katerih rezultat je odvisen od naključnih fizičnih procesov in psevdogeneratorje naključnih števil (angl. pseudo random number generator), ki uporabljajo matematične funkcije. Imenujemo jih tudi kriptografsko varni generatorji in so deterministični, saj ni mogoče ugotoviti rezultat naslednjega ustvarjenega bita v polinomskem času.

2.2 Tehnologija NFC in mobilne naprave

NFC (Near Field Communication) je brezžična tehnologija kratkega dosega, ki omogoča izvedbo brezkontaktne (angl. contactless) komunikacije med dvema napravama [6]. Nastala je kot nadgradnja že dodobra uveljavljene radiofrekvenčne tehnologije RFID (Radio Frequency IDentification), ki uporablja radijske valove (angl. radio waves) za vzpostavitev brezžične povezave in izmenjavo podatkov [8]. Komunikacija poteka preko elektromagnetnih valov, katerih doseg je odvisen od frekvence in lastnosti magnetnega polja. Najbolj pogoste frekvence RFID so:

- nizka frekvenca (Low Frequency – LF) – med 125 in 134 KHz z dosegom do 20 centimetrov,
- visoka frekvenca (High Frequency – HF) – 13.56 MHz z dosegom do največ 400 centimetrov in
- ultra visoka frekvenca (Ultra High Frequency – UHF) – med 400 in 930 MHz z dosegom do 100 metrov.

Sistem RFID sestavljajo čitalec (angl. reader) in odzivnik (angl. transponder). Prvi vsebuje oddajnik (angl. transceiver), kontrolno enoto in anteno, drugi pa anteno in vezje za hranjenje podatkov.

Tehnologija RFID se je začela uporabljati šele v 80. letih prejšnjega stoletja, čeprav njeni začetki segajo že med drugo svetovno vojno. Dandanes se uporablja za sledenje različnih proizvodov v trgovinah in skladiščih. Tako

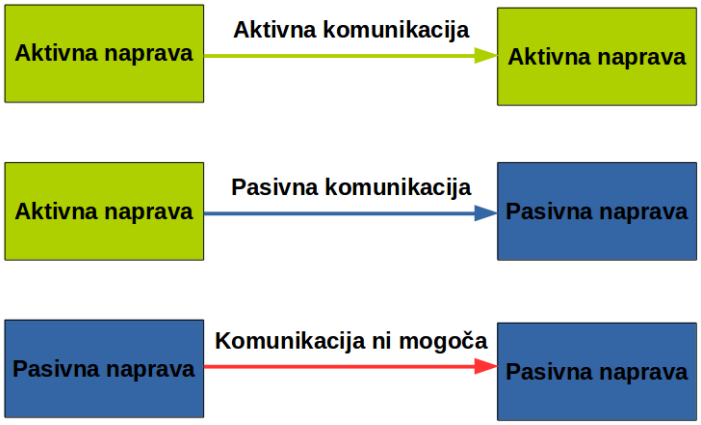
zamenjuje obstoječe označevanje izdelkov s črtno kodo. Z uporabo tehnologije RFID se zmanjšajo predvsem stroški skladiščenja in prevoza. Razlog za nadomeščanje črtnih kod z elektronskimi je tudi pohitritev in avtomatizacija nakupa izdelkov v trgovinah. Tako trgovcu ni potrebno branje črtne kode vsakega izdelka posebej, ampak lahko pridobi vse informacije o izdelkih naenkrat.

Tehnologija NFC je razširitev standarda RFID pri visoki frekvenci HF (13.56 MHz), a ima krajši doseg (le nekaj centimetrov). Z njo je mogoče prenašati podatke med dvema napravama na razdalji do 10 centimetrov. NFC je skladen z obstoječimi standardi RFID, kar mu omogoča branje podatkov z odzivnikov in tudi njihovo simuliranje. Razlikujeta se tudi v tem, da tehnologija NFC omogoča izvedbo dvosmerne (angl. bidirectional) komunikacije, RFID pa ne. Zaradi kratkega dosega je z NFC možno branje ene same značke (angl. tag). Pomembna značilnost tehnologije RFID je hkratno branje večjega števila značk.

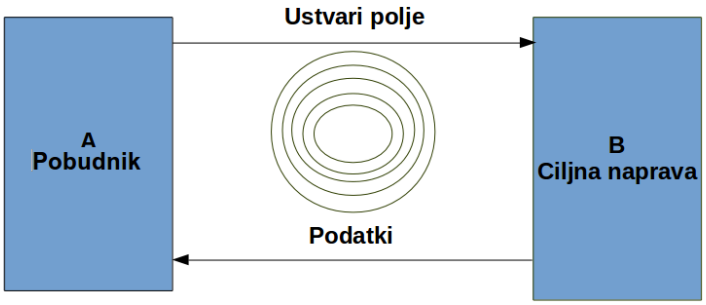
Naprava NFC lahko deluje v aktivnem ali pasivnem načinu. Aktivne naprave vsebujejo lasten vir energije za svoje napajanje in so sposobne ustvariti radiofrekvenčno polje. Iz slednjega pasivne naprave pridobijo energijo za napajanje. Glede na zgoraj omenjeno vrsto uporabljenih naprav, se komunikacija prav tako deli na aktivno in pasivno (slika 2.3). Med dvema aktivnima napravama poteka aktivna vrsta komunikacije, med aktivno in pasivno napravo poteka pasivna vrsta komunikacije, medtem ko komunikacija med dvema pasivnima napravama ni mogoča.

Naprave NFC se delijo tudi glede na njihovo vlogo v komunikaciji (slika 2.4). Naprava je bodisi pobudnik komunikacije (angl. initiator) bodisi ciljna naprava (angl. target device). Pobudnik komunikacije je lahko le aktivna naprava. Ciljna naprava pa je lahko pasivna ali aktivna.

Značka NFC je pasivna naprava namenjena hranjenju podatkov. Nima lastnega vira energije, zato ni sposobna ustvariti radiofrekvenčnega polja in začeti komunikacije. Sodeluje lahko le v komunikaciji z aktivno napravo.



Slika 2.3: Delitev komunikacije NFC glede na vrsto naprav NFC.



Slika 2.4: Delitev naprav NFC glede na njihovo vlogo v komunikaciji.

Čitalec NFC je aktivna naprava, ki je sposobna ustvariti dvosmerno povezavo z drugo, bodisi aktivno bodisi pasivno napravo NFC. Obstajajo zunanji in notranji čitalci NFC. Tipičen primer zunanjega čitalca je terminal PoS, ki ga uporabljajo v trgovinah in je namenjen brezkontaktnemu plačevanju. Notranji čitalci NFC so vgrajeni v druge naprave kot so pametne mobilne naprave ali telefoni ter tablice. Slednji so naprave, ki vključujejo čitalec in kartico ali značko v eni sami napravi.

Rešitve tehnologije NFC so standardizirane s strani naslednjih teles: ISO (International Organization for Standardization) in IEC (International Electrotechnical Commission) ter ETSI (European Telecommunications Standards Institute) in ECMA (European Computer Manufacturers Association). Protokol NFCIP-1 (Near Field Communication Interface and Protocol 1)

[18, 21] se nanaša na fizično plast tehnologije NFC in opisuje radiofrekvenčno polje, vmesnik, osnovni tok protokola ter transportni protokol. Prav tako določa obe vrsti komunikacije (aktivno in pasivno) ter definira naslednje tri načine delovanja (angl. operating mode):

- **Vsak z vsakim – P2P (angl. peer-to-peer)** način omogoča aktivno vrsto komunikacije med dvema napravama po protokolu NFCIP-1. Aktivna komunikacija je podprta še s pomočjo protokola LLCP (Logical Link Control Protocol) [27].

- **Bralno/pisalni (angl. reader/writer)** način uporablja pasivno vrsto komunikacije protokola NFCIP-1. Pri tem načinu mora aktivna naprava najprej ustvariti radiofrekvenčno polje, če želi prebrati informacije s pasivne naprave.

- **Emulacija kartice (angl. card emulation)** je način, kjer se naprava NFC obnaša kot brezkontaktna pametna kartica oziroma elektronska kartica, določena s standardom ISO/IEC 14443 [25] ali ISO/IEC 15693 [26]. Uporablja se pasivna vrsta komunikacije, saj so informacije lahko prebrane s pomočjo katerekoli naprave, ki je sposobna ustvariti radiofrekvenčno polje.

Protokol NFCIP-2 (Near Filed Communication Interface and Protocol 2) [19, 22] se uporablja za izbiro enega izmed načinov delovanja. Pomembno standardizacijsko telo je tudi organizacija NFC Forum [42], ustanovljena leta 2004. Gre za neprofitno združenje podjetij, katerih namen je razvoj novih standardnih rešitev tehnologije NFC in zagotavljanje združljivosti. Najbolj znani rešitvi sta protokol LLCP (Logical Link Control Protocol) in format zapisa podatkov NDEF (NFC Data Exchange Format) [28].

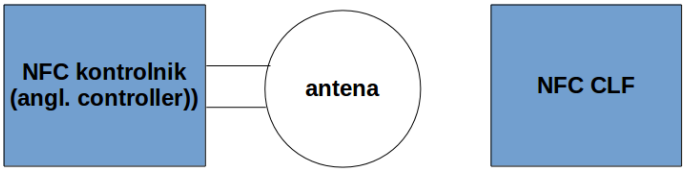
Tako, kot pri vseh preostalih tehnologijah je tudi NFC izpostavljena različnim varnostnim grožnjam. Pri vseh treh načinih delovanja obstaja možnost izvedbe različnih napadov in zlorab. Kratek doseg tehnologije ne zagotavlja popolne varnosti. Za njo je potrebo poskrbeti s standardnimi in

že uveljavljenimi varnostnimi mehanizmi. Sama stopnja varnosti je odvisna od vrste sistema in občutljivosti uporabljenih podatkov.

Vključitev tehnologije NFC v pametne mobilne naprave, nizka poraba energije in predvsem hitra povezava so razlogi za njeno vsepogostejšo uporabo na različnih področjih, kot so prenos podatkov med dvema mobilnima napravama NFC, brezkontaktno poslovanje, samodejen vklop funkcij mobilne naprave s pomočjo značke NFC (vklop in izklop WiFi-ja ali Bluetoth-a, predvajanje glasbe, vklop profila tihi način, nastavitve alarmov za bujenje, zagon različnih aplikacij itd.), avtomatizacija doma (prižiganje/ugašanje luči, odklepanje/zaklepanje vrat, prižig osebnega računalnika itd.), označevanje lokacij in nadomeščanje kode QR (Quick Response) na oglaševalskih plakatih.

2.2.1 Arhitektura mobilnih naprav NFC

Temeljne komponente mobilnih naprav, ki vključujejo tehnologijo NFC so vmesnik (angl. interface) NFC, varni element SE (Secure Element) in krmilnik gostitelja (angl. host controller) [6]. Na sliki 2.5 je prikazan vmesnik NFC, ki vključuje krmilnik (angl. controller) NFC, anteno NFC in brezkontaktni analogni/digitalni čelni del NFC CLF (NFC Contactless Front-end), s katerim ima uporabnik neposreden stik. Antena NFC je povezana s krmilnikom NFC, ki je namenjen komunikaciji z drugimi napravami NFC. Slednji omogoča pasivno in aktivno vrsto komunikacije. Skladen je s protokolom NFCIP-1 in vsemi tremi načini delovanja tehnologije NFC ter z ostalimi standardi RFID, kot sta ISO/IEC 14443 in ISO/IEC 15693.



Slika 2.5: Vmesnik NFC.

Varni element SE predstavlja varno in dinamično okolje namenjeno izvajanju aplikacij, ki uporabljajo tehnologijo NFC in njihovim podatkom. Sesta-

vljen je kot skupek oziroma množica različne strojne in programske opreme ter vmesnikov in protokolov, vključenih v samo mobilno napravo. Vsaka mobilna naprava z vključeno tehnologijo NFC mora imeti vsaj en varni element. Lahko jih ima tudi več. Za povezavo varnega elementa s krmilnikom NFC sta na voljo standardizirana fizična vmesnika SWP (Single Wire Protocol) [23] in NFC-WI (NFC Wired Interface) [20]. SWP določa eno žično povezavo med varnim elementom in krmilnikom NFC na principu delovanja, imenovanem nadrejeni – podrejeni (angl. master/slave). Nadrejenega v komunikaciji predstavlja krmilnik NFC, podrejenega pa varni element. NFC-WI določa dvožično povezavo, kjer je ena žica namenjena vhodnemu, druga pa izhodnemu signalu. Uporabljajo se naslednje izvedbe varnega elementa:

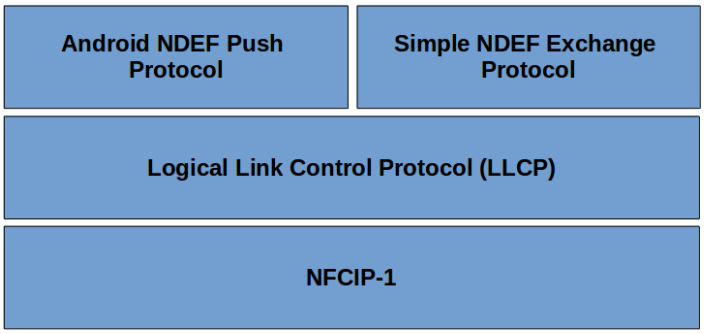
- **Vgrajena (angl. embedded) strojna oprema** je neodstranljiv del oziroma čip mobilne naprave. Ima veliko stopnjo varnosti, saj je v mobilno napravo vgrajen že pri proizvajalcu.
- **SMC (Secure Memory Card)** je odstranljiva kartica, ki združuje pomnilniško in pametno kartico. Njena prednost je predvsem prenosljivost med mobilnimi napravami in velikost pomnilnika.
- **UICC (Universal Integrated Circuit Card)** kartica pa je fizična pametna kartica na kateri temelji tudi SIM (Subscriber Identity Module).

Pomemben del mobilne naprave NFC je tudi krmilnik gostitelja HC (Host Controler), ki je namenjen vzpostavitvi povezave med krmilnikom NFC in varnim elementom ter obdelavi podatkov. Povezavo med njim in krmilnikom NFC omogoča vmesnik HCI (Host Controller Interface). Njegova uporaba omogoča krmilniku NFC komunikacijo z aplikacijo in z enim ali več varnih elementov.

2.2.2 Komunikacija vsak z vsakim (P2P)

Komunikacija vsak z vsakim (v nadaljevanju P2P) je realizirana s pomočjo svežnja med seboj odvisnih standardiziranih protokolov, ki so zgrajeni en nad

drugim (slika 2.6) [6]. P2P je dvosmerna komunikacija med dvema aktivnima napravama NFC, kjer ena oddaja podatke, druga pa jih sprejema. Slednja lahko prične z oddajanjem podatkov šele, ko ga prva zaključi. Odločitev katera naprava bo oddajala in katera sprejemala je sprejeta po opravljenem začetnem rokovanju (angl. initial handshake) in je odvisna od same implementacije aplikacije.



Slika 2.6: Standardizirani protokoli v izvedbi komunikacije P2P.

Osnovno izvedbo komunikacije P2P določa protokol NFCIP-1, kot komunikacijo dveh aktivnih naprav NFC – aktivni način komunikacije. Slednji omogoča obema sodelujočima napravama izmenično ustvarjanje radiofrekvenčnega polja. Sestavljen je iz inicializacije in transportnega protokola. Izmenično ustvarjanje radiofrekvenčnega polja zahteva v postopku inicializacije uporabo protokola namenjenega izogibanju kolizijam (angl. collision avoidance). Ta zagotavlja, da naprava NFC ne more ustvariti radiofrekvenčnega polja, če je v njeni bližini že prisotno katero drugo polje.

Nad protokolom NFCIP-1 je zgrajen protokol LLCP (Logical Link Control Protocol) [27], namenjen izboljšanju funkcionalnosti prvega. Uvrščen je med podatkovno-povezavne (angl. data link) protokole in standardizira naslednje funkcionalnosti:

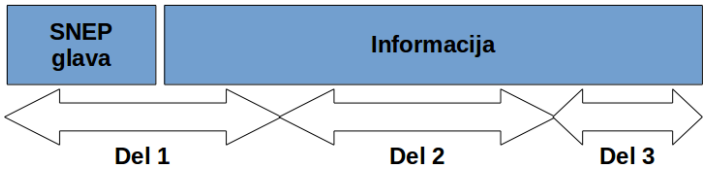
- brezpovezavni (angl. connectionless) prenos, ki omogoča spontano izmenjavo podatkovnih enot, vendar ne zagotavlja uspešnega prenosa in nima kontrole toka podatkov,

- povezavni (angl. connection-oriented) prenos, ki omogoča urejen prenos podatkov in njihovo zanesljivo dostavo,
- vzpostavitev (angl. establishment), nadzor (angl. control) in prekinitev povezave ter
- asinhrono (angl. asynchronous) komunikacijo.

Na aplikacijskem nivoju tehnologije NFC (nad protokolom LLCP) sta uporabljena protokola NPP (Android NDEF Push Protocol) [24] in SNEP (NFC Forum Simple NDEF Exchange Protocol) [30]. Omogočata prenos sporočil zapisanih v formatu NDEF. NPP je bil razvit kot enosmerni protokol (angl. one way), SNEP pa je bil zasnovan na principu zahteva – odgovor (angl. request/response) princip. Gre za bolj napreden protokol, standardiziran s strani organizacije NFC Forum. V komunikaciji sta prisotni dve napravi NFC, pri kateri eno imenujemo odjemalec (angl. client) SNEP, drugo pa strežnik (angl. server) SNEP. Vsako sporočilo SNEP sestoji iz glave in informacije. Glava vsebuje naslednja tri polja:

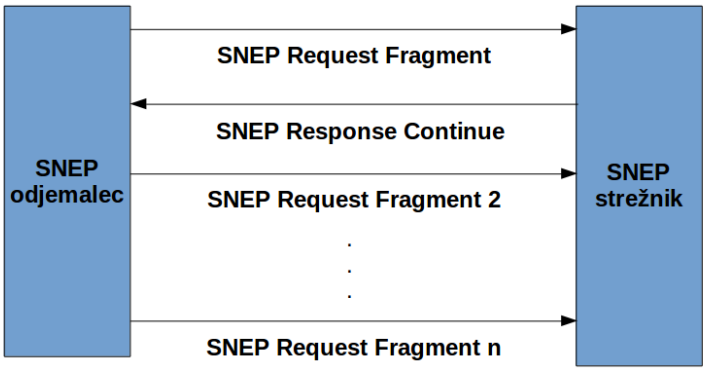
- različica velikosti enega bajta,
- vrsta sporočila (Put, Get, Request itd.) velikosti enega bajta in
- dolžina velikosti štirih bajtov.

Informacijo predstavlja sporočilo NDEF, katerega velikost je omejena na $2^{32}-1$ in predstavlja največjo celoštevilsko vrednost, ki je lahko še zapisana v polju dolžina. Zato se večja sporočila prenašajo po delih (angl. fragment), kot prikazuje slika 2.7.



Slika 2.7: Primer sporočila SNEP, ki je razdeljeno na tri dele.

Naprava, ki prejme prvi del sporočila mora sporočiti drugi napravi ali je sposobna prejeti tudi ostale dele sporočila. Pošiljanje vseh delov sporočila je omogočeno le takrat, ko pošiljatelj prejme potrditev za nadaljnje pošiljanje. Shematičen postopek izmenjave sporočila po delih je predstavljen na sliki 2.8.

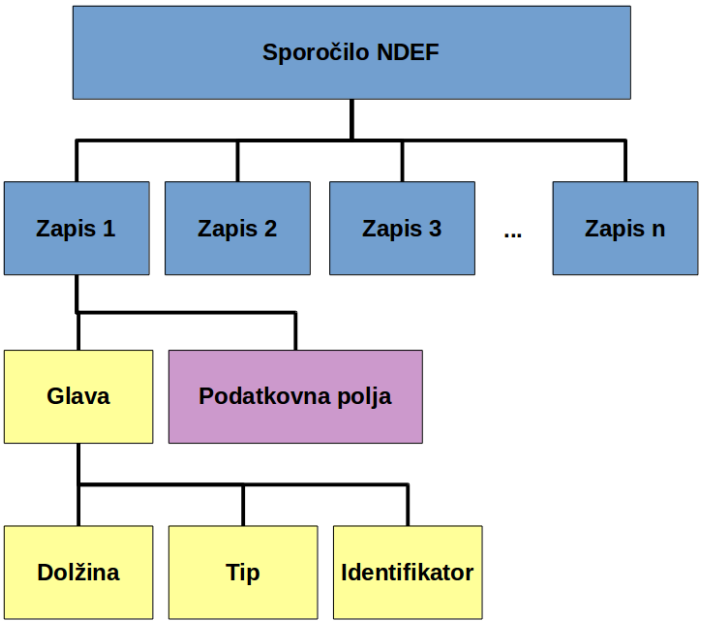


Slika 2.8: Izmenjava sporočila po delih med odjemalcem in strežnikom SNEP.

2.2.2.1 Format podatkov NDEF

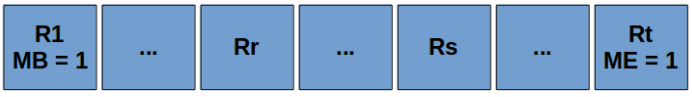
Format NDEF je standardiziran (organizacija NFC Forum) način zapisa podatkov v binarni obliki, ki združuje več koristne vsebine (angl. payload) [6]. Vsako sporočilo NDEF je sestavljeno iz enega ali več zapisov (angl. record), vsak zapis pa iz glave in enega ali več podatkovnih polj (slika 2.9). Glavo zapisa sestavljajo tri polja, ki opisujejo podatkovna polja v zapisu s pomočjo naslednjih parametrov:

- dolžina (angl. payload length), ki določa število podatkovnih polj,
- tip (angl. payload type), ki opisuje vrsto podatkovnih polj in
- identifikator (angl. payload identifier), ki je opcijski in zapisan v absolutni obliki URI (Uniform Resource Identifier).



Slika 2.9: Sestava sporočila NDEF.

Slika 2.10 predstavlja format sporočila NDEF v katerem prvi zapis vsebuje zastavico MB (Message Begin), zadnji pa ME (Message Ends). Ko sta njuni vrednosti nastavljeni na 1, določata začetek in konec sporočila. Sporočilo mora vsebovati minimalno en zapis. Glava sporočila NDEF se nahaja na levi, rep pa na desni strani sporočila ($r < s < t$) (slika 2.10).



Slika 2.10: Format sporočila NDEF.

Število zapisov v sporočilu NDEF je neomejeno, zato se v primeru večjih podatkov uporablja veriga zapisov. Podatki se dinamično razdelijo na več zapisov različnih vrst. Začetni zapis vsebuje zastavico CF (Chunk Flag) postavljeno na ena. V njem je tudi zahtevana definicija dolžine celotnih podatkov, tipa in identifikatorja, ki se ne spreminja do konca prenosa vseh zapisov. Srednji zapisi imajo prav tako zastavico CF postavljeno na ena, kar pove, da imajo enak tip podatkov z istim identifikatorjem kot prvi zapis.

Dolžina v tem primeru predstavlja le dolžino trenutnega zapisa. Končni zapis nima postavljene zastavice CF na ena, kar določa zadnji zapis z istim tipom podatkov in identifikatorjem kot tisti pred njim.

Zapis NDEF vsebuje tudi 3-bitno polje TNF (Type Name Format), ki nakazuje strukturo podatkovnih polj. Organizacija NFC Forum je standardizirala kar nekaj tipov RTD (Record Type Definition) za sporočila NDEF [29]. Vsako polje TNF vsebuje niz, ki ga uporabljajo aplikacije za določanje strukture vsebine zapisa. Imena so zapisana s pomočjo absolutne reference URI. Tipi so razdeljeni na naslednji dve skupini:

- Tip NFC Forum Well-Known z imenskim področjem “nfc” in “wkt”. (primer: urn:nfc:wkt:complicated-type)
  Skupina se dodatno deli še na: globalni (angl. global) tip, ki se začne z veliko začetnico in lokalni (angl. local) tip, ki se začne z majhno začetnico.

- NFC Forum External z imenskim področjem ”nfc” ter ”ext”. (primer: urn:nfc:ext:test)

Med najpogostejše tipe RTD spadajo pametni posterji, naslovi URI, digitalni podpisi in tekst. Tip URI RTD se uporablja za prenos zapisa URI iz ene naprave NFC na drugo ali pa za pridobitev zapisa URI iz značke NFC. Polje URI je definirano s pomočjo kodiranja UTF-8. Tekstovni tip RTD vsebuje čistopis v prosti obliki, kodiran s pomočjo zapisa UTF-8 ali UTF-16.

Izmenjava podatkov zajetih v sporočilih NDEF zahteva zanesljiv transportni protokol. Uporablja se že zgoraj opisani protokol LLCP, ki omogoča zaporeden povezavno orientiran prenos in skrbi za njegovo uspešno izvedbo.

2.2.2.2 Varnost

Tehnologija NFC je iz osnovnih karakteristik RFID prevzela tudi njene probleme, predvsem s področja varnosti in zasebnosti [9]. Osnovni protokol NFCIP-1, na katerem temelji celotna komunikacija NFC, ne zagotavlja ni-

kakršnih varnostnih mehanizmov. Posledično se tehnologija NFC v načinu delovanja P2P srečuje z naslednjimi grožnjami in napadi [6, 16]:

- **Prisluškovanje (angl. eavesdropping)** je primer, kjer napadalec uporabi čitalec z namenom, da posname komunikacijo med dvema avtoriziranima napravama NFC. Razdalja pri kateri je napad še mogoč znaša od 10 centimetrov do tudi 10 metrov. Določitev točne razdalje ni možna, zaradi odvisnosti od prevelikega števila parametrov (vir energije, lokacija, karakteristike radiofrekvenčnega polja, kvaliteta napadalčeve antene in dekodirnika radiofrekvenčnega signala itd.).

- **Uničenje (angl. corruption), spreminjanje (angl. modification) in vstavljanje (angl. insertion) podatkov**. Namen uničenja podatkov je predvsem zmotiti napravo NFC tako, da ta ne razume prejetih podatkov. Slednje lahko napadalec doseže s poznavanjem uporabljene modulacije in kodiranja. Takšne vrste napad se imenuje tudi napad s preobremenitvijo ali napad DoS (Denial of Service). Če želi napadalec vstaviti podatke, mora to storiti pred pošiljanjem izvirnih podatkov. Takšen način napada je možen pri dolgotrajnejših prenosih, sicer samo vstavljanje ni uspešno in se podatki uničijo.

- **MITM (Man-in-The-Middle)** je napad, kjer se v komunikacijo dveh avtoriziranih, vrine še tretja neavtorizirana naprava NFC. Tako celotna komunikacija poteka preko neavtorizirane naprave NFC.

- **Kraja (angl. stolen)** mobilne naprave NFC je splošen napad, ki ni povezan s komunikacijo P2P. Napadalec ukrade mobilno napravo NFC z namenom oškodovanja uporabnika (prodaja naprave, uporaba mobilnih storitev, uporaba aplikacij/storitev za plačevanje itd.).

Zaradi kratkega dosega tehnologije NFC je zgornje napade zelo težko izpeljati. Razlog je predvsem njihova nepraktičnost in neučinkovitost. Kljub temu morajo za varnost poskrbeti aplikacije, zgrajene nad tehnologijo NFC.

Komunikacijo P2P lahko pred zgornjimi napadi zaščitimo predvsem z uporabo šifriranja in varnih kanalov. Uporabo ukradene mobilne naprave lahko preprečimo z različnimi načini avtentikacije, ki onemogoča neavtorizirani osebi dostop do podatkov in storitev.

2.2.3 Android in NFC

Operacijski sistem Android je odprtokodna platforma namenjena mobilnim napravam, kot so pametni mobilni telefoni in tablice [7]. Naprave so opremljene z zaslonom na dotik in imajo manjšo računsko zmogljivost. Jedro operacijskega sistema temelji na Linuxu (različica 2.x ali 3.x), ki je vmesna plast med strojno ter programsko opremo. Skrbi predvsem za upravljanje s pomnilnikom, procesi, omrežnim skladom in gonilniki.
Za razvoj aplikacij je uporabljen programski jezik Java. Javanska programska koda v datotekah *.java* se s pomočjo standardnega prevajalnika javac prevede v bajtno kodo (angl. byte code), zapisano v datotekah *.class*. Sledi pretvorba v izvršljive datoteke *.dex*, imenovane Davlik, ki se izvajajo na virtualni napravi imenovani DVM (Davlik Virtual Machine). Slednje so nato, s pomočjo orodja za arhiviranje, zapakirane v Android paket (*.apk*), ki omogoča namestitev aplikacije na mobilni napravi.
DVM je glavni del zagonskega okolja, zasnovan posebej za Android naprave, ker omogoča optimizacijo navadnih javanskih zagonskih datotek. Optimizacija je potrebna, ker imajo Android naprave omejeno računsko zmogljivost, velikost pomnilnika in napajanje (uporaba baterije). Vsaka aplikacija se izvaja v svojem procesu (instanca DVM), imenovanem peskovnik (angl. sandbox), kar jih varno loči od sistema in preostalih aplikacij.
Operacijski sistem vsebuje množico osnovnih knjižnic, ki se deloma prekrivajo s standardno knjižnico Jave (Java SE Library). Poleg osnovnih knjižnic je razvijalcem na voljo knjižnica Android API, sestavljena iz različnih paketov, ki omogočajo učinkovito uporabo različnih Android komponent, funkcij in storitev.

2.2.3.1 Komponente aplikacij

Android aplikacije so lahko sestavljene iz naslednjih komponent [7]:

- **aktivnost (angl. activity)**,
- **storitev (angl. service)**,
- **prejemnik sporočil (angl. broadcast reciever)** in
- **upravitelj vsebine (angl. content provider)**.

Aktivnost predstavlja posamezni zaslon aplikacije in omogoča njeno izvajanje z vključenim uporabniškim vmesnikom. Vsaka aplikacija mora imeti vsaj eno aktivnost. Če jih ima več, so aktivnosti med seboj ločene. Do posamezne aktivnosti lahko uporabnik dostopi preko vnaprej določenega postopka. Aktivnosti lahko prožijo različne komponente aplikacije in druge aplikacije. Med preklapljanjem aktivnosti se notranje stanje trenutne aktivnosti shrani za njeno kasnejše izvajanje. Hranjenje se preneha v primeru pomanjkanja pomnilnika. Vsaka aktivnost je lahko v enem izmed naslednjih stanj:

- aktivna (angl. active),
- prekinjena (angl. paused),
- ustavljena (angl. stopped) in
- uničena (angl. destroyed).

Aktivna se nahaja v ospredju aplikacije, uporabniku je prikazana na zaslonu in se odziva na njegove akcije. Prekinjena aktivnost je le delno vidna na zaslonu (v ozadju), saj je njeno izvajanje začasno ustavljeno. Razlogi za prekinitev izvajanja aktivnosti so različni dialogi, izbire in potrjevanja. Ustavljena aktivnost uporabniku ni vidna, saj se njeno notranje stanje shrani za kasnejše izvajanje. V primeru pomanjkanja pomnilnika se hranjenje zavrže in aktivnost je uničena.

Storitev omogoča izvajanje različnih nalog v ozadju, ko ne vključujejo uporabniškega vmesnika. Na voljo sta dve vrsti storitev:

- začeta (angl. started) in
- vezana (angl. bound).

Izvajanje začete storitve prične ena izmed aplikacijskih komponent. Izvajati se začne v ozadju dokler se ne zaključi tudi, če je aplikacijska komponenta, ki jo je začela medtem uničena. Vezana storitev omogoča interakcijo s samo aplikacijo, saj omogoča pošiljanje zahtevkov in rezultatov.

Prejemniki sporočil so aplikacijske komponente, namenjene sprejemanju različnih obvestil operacijskega sistema (sprememba časovnega pasa, status baterije, zaprtje zaslona itd.). Ne vključujejo uporabniškega vmesnika, vendar omogočajo prikaz različnih obvestil v statusni vrstici in ustrezno prilagoditev aplikacije glede na prejeta obvestila.

Upravitelj vsebine skrbi za aplikacijske podatke. Omogoča pridobivanje potrebnih podatkov in njihovo shranjevanje. Z njegovo pomočjo si lahko aplikacije med seboj delijo različne podatke.

2.2.3.2 NFC

Za razvoj aplikacij, ki uporabljajo tehnologijo NFC sta na voljo dva paketa. Osnovi $android.nfc$ je na voljo od različice 9 (Android 2.3, 2.3.1 in 2.3.2) naprej in omogoča pisanje ter branje sporočil NDEF z značke in izvedbo izmenjave podatkov med dvema mobilnima napravama NFC. Paket sestavljajo naslednji razredi:

- Tag, ki predstavlja značko,
- $NfcAdapter$, ki predstavlja vmesnik NFC na mobilni napravi,
- $NfcManager$ za pridobitev instance vmesnika NFC,
- $NdefMessage$, ki predstavlja sporočilo NDEF in

- *NdefRecord*, ki predstavlja zapis NDEF.

Paket *android.nfc.tech* je na voljo od različice 10 (Android 2.3.3 in 2.3.4) naprej in omogoča uporabo različnih vrst značk (IsoDep, Mirafare, NFC-A, NFC-B, NFC-F, NFC-V itd.). Vključuje nabor razredov za izvajanje različnih operacij, predvsem za pisanje in branje podatkov različnih formatov.

Pomemben del predstavlja tudi sistem za razporejanje značk (angl. tag intent dispatch system), ki deluje le kadar naprava ni zaklenjena. Namenjen je zagonu aplikacije s pomočjo vnaprej določenih značk. Vsaki aplikaciji, ki uporablja tehnologijo NFC, je potrebno določiti s katerimi značkami lahko upravlja. Naloga sistema za razporejanje je izbrati aplikacijo, ki bo znala rokovati z zaznano značko. V primeru določitve značke več kot eni aplikaciji, sistem prikaže meni, kjer uporabnik izbere ustrezno aplikacijo. Če značka ni določena nobeni aplikaciji, potem sistem sam poišče splošno aplikacijo, ki je namenjena obdelavi širše skupine značk. Sistem ugotovi tip značke na podlagi vsebine prvega zapisa sporočila NDEF. Uspešnost pravilnega razporejanja je odvisna od natančne določitve tipa značk. Sistem za razporejanje značk deluje na naslednji način:

- Ugotovitev tipa značke z branjem značke.
- Enkapsulacija tipa in vsebine v razred *Intent*.
- Zagon ustrezne aktivnosti aplikacije, katera je določena za obdelavo značke.

Tip značk, ki jih bo aplikacija lahko obdelala, je mogoče določiti v datoteki *AndroidManifest.xml* s pomočjo filtrov namer (angl. intent filter). Posamezna aplikacija lahko uporablja svoje lastne tipe, ki so namenjeni le njej. Filter namer se doda posamezni aktivnosti, ki se odpre, če sistem zazna značko, katere tip je določen v tej aktivnosti. Za lažjo določitev aplikacije za obdelavo podatkov značke, je v njem potrebno določiti prioriteto:

- ACTION_NDEF_DISCOVERED ima najvišjo prioriteto. Če značka vsebuje podatke v formatu NDEF, potem sistem skuša zagnati aktivnost aplikacije z določenim tipom značke in to prioriteto.

- Če značka vsebuje podatke v formatu NDEF, vendar nobena aplikacija ni določena za obdelavo njenega tipa, potem sistem ustvari prioriteto ACTION_TECH_DISCOVERED in skuša zagnati aktivnost s takšnim tipom in prioriteto.

- Če nobena aplikacija nima za obdelavo določenega tipa, ki je na znački in ene od zgornjih dveh prioritet, potem sistem ustvari najnižjo prioriteto ACTION_TAG_DISCOVERED.

Izmenjava sporočil NDEF med dvema Android napravama z vključeno tehnologijo NFC se imenuje žarek (angl. beam), realiziran s pomočjo funkcije Android Beam. Napravi, ki želita komunicirati, morata podpirati ali protokol NPP ali SNEP. Pri uporabi funkcije Android Beam mora aplikacija, ki želi poslati sporočilo, delovati v ozadju, obe napravi pa morata imeti odklenjen zaslon. Iz vidika varnosti je ta zahteva uporabna, saj se tako oba uporabnika naprav zavedata same komunikacije.
Naprave z minimalno različico 14 (Android 4.0, 4.0.1 in 4.0.2) ali več lahko nemoteno uporabljajo vse podprte funkcionalnosti tehnologije NFC. Vsi potrebni paketi za delo s tehnologijo NFC, so v celoti na voljo od različice 16 naprej.

2.2.3.3 Varno hranjenje podatkov

Idealno rešitev za varno izvajanje aplikacije in hranjenje njenih podatkov predstavlja varni element [6]. Slednji predstavlja del strojne opreme mobilne naprave z vključeno tehnologijo NFC, katerega stanje ni mogoče spreminjati. Na mobilnih napravah Android je uporaba vgrajenega varnega elementa omejena in ni podprta, zato tudi ni na voljo nobenega uradnega paketa (API). Trenutno je uporaba vgrajenega elementa omogočena le proizvajalcem mobilnih naprav z vključeno tehnologijo NFC. Dostop do kartice UICC oziroma SIM, ki se lahko uporablja kot varni element je odvisna od ponudnikov telekomunikacijskih in mobilnih storitev. Prav tako obstaja veliko mobilnih naprav NFC, predvsem tablic, ki ne podpirajo tovrstnih kartic. Implementacija var-

nega elementa je mogoča tudi na odstranljivih karticah SMC, vendar njihovo uporabo in delovanje ne omogočajo vse naprave. Zaradi nepraktičnosti in omenjenih omejitev varnega elementa, je za varno hranjenje zasebnih podatkov treba poiskati druge rešitve in sredstva, ki jih ponuja platforma Android.

Za izolacijo podatkov aplikacije in ločeno izvajanje programske kode od preostalih aplikacij na mobilnih napravah Android, poskrbi peskovnik [43]. Dostopnost do zasebnih podatkov aplikacije pa je tudi odvisna od izbrane možnosti hranjenja [44]:

- deljene nastavitve (angl. shared prefrences),
- notranja shramba (angl. internal storage),
- podatkovna baza SQLite (angl. SQLite database) in
- zunanja shramba (angl. external storage).

Zasebni podatki shranjeni v deljenih nastavitvah so lahko le primitivnih tipov (cela števila, nizi, decimalna števila itd.) na osnovi parov ključ – vrednost (angl. key/value). Dostop do njih je mogoč le znotraj same aplikacije, kateri so namenjene. Notranja shramba omogoča hranjenje zasebnih podatkov v pomnilniku mobilne naprave. Podatki aplikacije shranjeni v pomnilniku naprave so namenjeni zgolj tej aplikaciji tako, da druge naprave do njih nimajo dostopa. Z odstranitvijo aplikacije se odstranijo tudi podatki znotraj pomnilnika naprave. Uporaba podatkovne baze SQLite omogoča strukturirano hrambo zasebnih podatkov aplikacije tako, da so ti dostopni kjerkoli znotraj aplikacije, preostale aplikacije pa do njih prav tako nimajo dostopa. Z razliko od preostalih treh možnosti hranjenja podatkov, si podatke v zunanji shrambi aplikacije lahko delijo med seboj. Vsaka aplikacija, ki je nameščena na napravi lahko v njo zapisuje podatke in jih iz nje bere. Med zunanje shrambe spadajo različni mediji, kot so odstranljivi in tudi notranji neodstranljivi podatkovni pomnilniki.

Ob uporabi zgornjih načinov hranjenja zasebnih podatkov, mora aplikacija poskrbeti za njihovo varnost. Čeprav operacijski sistem Android omejuje

in nadzoruje dostop do podatkov, ti niso povsem varni. Podatke je še vedno mogoče pridobiti s pomočjo fizičnega dostopa do pomnilnika ali popolnoma odklenjene (angl. rooted) mobilne naprave. Dodatno zaščito zasebnih podatkov predstavlja simetrično šifriranje, ki za šifriranje in dešifriranje uporablja isti ključ. Hranjenje ključa v programski kodi (angl. hardcode) ali na sami napravi predstavlja težavo, saj ga je mogoče enostavno pridobiti s pomočjo podrobnega pregledovanja (angl. reverse engineering) ali popolnoma odklenjene naprave. Izognitev hranjenju ključa na mobilni napravi omogoča izpeljava (angl. derive) ključa s pomočjo gesla, ki ga uporabnik vnese ob prijavi v aplikacijo.

2.2.4 Mobilno plačevanje

Priljubljenost in razširjenost tehnologije NFC (vedno večji delež pametnih mobilnih naprav z vgrajenim NFC) sta pripomogli k razvoju številnih komercialnih (Google Wallet, Apple Pay, Visa PayWave itd.) in raziskovalnih (IDA-Pay) sistemov za brezkontaktno mobilno plačevanje. Tovrstni sistemi delujejo na principu plačevanja z obstoječimi plačilnimi karticami in terminali PoS. Mobilne naprave NFC hranijo informacije o eni ali več karticah, kar jim omogoča, da se obnašajo kot brezkontaktne plačilne kartice.

IDA-Pay je raziskovalna rešitev na področju mobilnega plačevanja [11]. Uporablja mobilne naprave Android z vgrajeno tehnologijo NFC v povezavi s terminali PoS. Informacija o plačilni kartici je varno shranjena v pomnilniku naprave s pomočjo šifriranja. Za pošiljanje informacij o plačilu med napravo in terminalom se uporablja način delovanja P2P tehnologije NFC. Celoten sistem je sestavljen iz naslednjih treh delov:

- Aplikacija IDA-Pay predstavlja odjemalca in je nameščena na pametni napravi Android z vgrajeno tehnologijo NFC.
- Aplikacija IDA-Pay PoS je povezana s strežnikom na internetu in vsebuje vmesnik NFC. Namenjena je izmenjavi podatkov o plačilu in

plačilni kartici z odjemalcem preko terminala PoS.

- Spletni strežnik IDA-Pay Gateway je namenjen posredovanju zahtevka o plačilu ustreznemu omrežju za delo s plačilnimi karticami.

Google Wallet je brezplačna storitev, ki omogoča povezano mobilno plačevanje v trgovinah v povezavi s terminalom PoS [35]. Sistem omogoča tudi plačevanje preko spleta z uporabo Googlove pošte Gmail. Uporaba storitve Google Wallet je trenutno mogoča le v Združenih državah Amerike. Glavni del sistema je aplikacija za Android mobilne naprave z vgrajeno tehnologijo NFC. Slednja deluje na osnovi emulacije plačilnih kartic na napravi. Aplikacija omogoča dodajanje različnih kreditnih in debetnih kartic ter promocijskih kod ali kuponov v mobilno napravo, katerih informacije se varno hranijo na mobilni napravi in v Googlovem oblaku. Varnost hranjenja informacij na napravi je mogoča z uporabo šifriranja in varnega elementa. Plačilo poteka tako, da uporabnik približa mobilno napravo NFC terminalu PoS in izbere vir plačila, če ima na napravi dodanih več različnih plačilnih kartic. Dodatno je aplikacija zavarovana s kodo PIN (Personal Idetification Number), ki jo mora uporabnik vnesti ob zagonu aplikacije.

Apple Pay je najnovejša storitev za mobilno plačevanje, ki jo je razvilo podjetje Apple [34]. Na voljo je na njihovi najnovejši mobilni napravi Apple iPhone 6, saj ima šele ta vključeno tehnologijo NFC. Plačilo se izvede tako, da uporabnik priloži napravo ob brezkontaktni čitalec (terminal PoS) medtem, ko ima svoj prst položen na znak TouchID izrisan na zaslonu. TouchID se nanaša na avtentikacijo uporabnika s pomočjo prstnih odtisov. Naprava nadomesti uporabo plačilne kartice. Uporabniku je omogočeno dodajanje več različnih kreditnih in debetnih kartic, katerim se dodeli unikatna številka računa DAN (Device Account Number). Njeno varno hranjenje je zagotovljeno s pomočjo šifriranja in uporabe varnega elementa.

Visa payWave je sistem plačevanja, ki že od leta 2007 naprej uporablja brezkontaktne plačilne kartice v povezavi s terminalom PoS [45]. Delovanje kartic temelji na tehnologiji RFID, zato se plačilo izvede brez njihovega vstavljanja v sam terminal. Izvedba je mogoča že na razdalji do 10 centimetrov od terminala PoS. Transakcije z manjšimi zneski se izvedejo takoj, sicer je potrebna avtentikacija z vnosom osebne identifikacijske številke PIN. Sistem omogoča tudi uporabo mobilnih naprav NFC, ki omogočajo emulacijo plačilnih kartic. Tako se naprava obnaša kot brezkontaktna kartica, kjer postopek plačila ostane enak.

Poglavje 3

Zasnova in načrtovanje

Zasnova sistema za brezkontaktno plačevanje z mobilnimi napravami NFC vključuje osnovni koncept in vrste uporabnikov, ki v njem sodelujejo. Podrobno so predstavljeni vsi glavni postopki, ki omogočajo uporabo elektronske gotovine. Arhitektura sistema je razdeljena na bančni sistem (strežniška aplikacija) in uporabniški del (mobilna aplikacija). Prva je namenjena testiranju in evalvaciji sistema ter omogoča njegovo celovito izvedbo. Druga pa predstavlja glavni del sistema in omogoča izvedbo brezkontaktnega plačila s pomočjo tehnologije NFC. Za vsak del sistema so posebej opisane funkcionalnosti in podatkovni model.

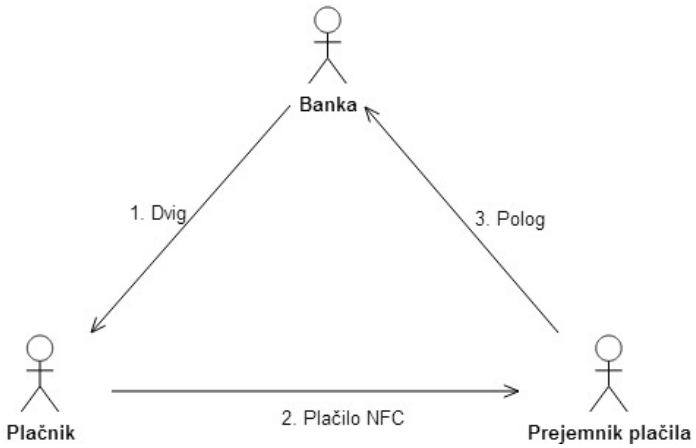
3.1 Brezkontaktni sistem plačevanja

Banka poleg običajnih storitev nudi tudi možnost uporabe sistema za brezkontaktno plačevanje z mobilnimi napravami NFC. Oseba, ki želi uporabljati tovrstno storitev mora biti komitent banke, kjer ima že odprt transakcijski račun (tR). Komitent lahko začne uporabljati sistem po opravljenem postopku za odprtje e-računa (eR) na mobilni napravi, ki ga opravi v banki. Postopek zajema namestitev ustrezne aplikacije na komitentovo mobilno napravo NFC in se izvede na osnovi interakcije med komitentom in bančnim uslužbencem ter med mobilno napravo NFC in bančnim sistemom.

Po končanem postopku lahko komitent na podlagi transakcijskega računa (tR), e-računa (eR) na mobilni napravi NFC in ustrezne aplikacije uporablja sistem za brezkontaktno plačevanje. Sistem temelji na uporabi elektronske gotovine. Z njo lahko plačuje le oseba, ki je izvedla dvig. To pomeni, da elektronska gotovina ni prenosljiva. Oseba, ki je prejela plačilo v obliki elektronske gotovine, lahko slednjo unovči le s pomočjo pologa, z njo pa ne more plačevati naprej. Zato je e-račun na mobilni napravi razdeljen na dva dela. V enem se hrani dvignjena elektronska gotovina, v drugem pa prejeta.

3.1.1 Osnovni koncept

Brezkontaktni sistem za plačevanje vključuje banko, plačnika in prejemnika plačila. Glavni del sistema je plačilo, ki se izvede s tehnologijo NFC na osnovi interakcije med plačnikom in prejemnikom plačila. Model sistema za brezkontaktno plačevanje z mobilnimi napravami NFC je prikazan na sliki 3.1. Plačnik bo pridobil elektronsko gotovino z ustreznim postopkom za dvig (1). Če želi plačnik od prejemnika plačila kupiti izdelek ali storitev, mu bo plačal z veljavno elektronsko gotovino (2). Prejemnik plačila bo pridobljeno elektronsko gotovino unovčil na banki s postopkom za polog (3).



Slika 3.1: Model brezkontaktnega plačila.

Elektronska gotovina se s pomočjo postopka za dvig prenese iz transak-

cijskega računa (tR) plačnika na njegov e-račun (eR). Plačnik jo bo med postopkom plačila prenesel iz svojega e-računa (eR) na prejemnikovega. Prejemnik plačila jo s postopkom za polog prenese iz svojega e-računa (eR) na transakcijski račun (tR). Tok elektronske gotovine je enak toku gotovine, kjer je e-račun predstavljen s fizično obliko denarnice.

Banka za plačnika in prejemnika plačila predstavlja zaupanja vredno tretjo stran. Njena vloga je skrb za izvedbo dviga in pologa elektronske gotovine. Oba postopka se izvedeta kot interakcija med banko in plačnikom ali med banko in prejemnikom plačila. Če želi plačnik dvigniti določeno vsoto elektronske gotovine, mora imeti na svojem transakcijskem računu (tR) dovolj finančnih sredstev, banka pa mora poskrbeti za ustrezen format elektronske gotovine in njeno veljavnost. Po uspešno zaključnem dvigu, banka jamči, da bo sprejela in prenesla ustrezen znesek na transakcijski račun (tR) prejemnika plačila, ko bo položil veljavno elektronsko gotovino. V postopku pologa je njena vloga tudi preverjanje veljavnosti elektronske gotovine s pomočjo njenega javnega ključa in možnosti izvedbe dvojne porabe.

Plačnik predstavlja osebo, ki plača določen izdelek ali storitev z elektronsko gotovino. Na svoji mobilni napravi NFC ima nameščeno aplikacijo za brezkontaktno plačevanje. Če želi izvesti plačilo, mora imeti zadostno količino elektronske gotovine, ki jo pridobi s pomočjo postopka za dvig. Željen znesek dviga posreduje banki, ki preveri stanje na njegovem transakcijskem računu (tR). Če ima dovolj sredstev, lahko nadaljuje s postopkom za dvig. Po zaključenem dvigu se stanje na njegovem transakcijskem računu (tR) zmanjša, pridobljena elektronska gotovina pa se varno shrani na njegovi napravi. Torej se stanje na njegovem e-računu (eR) poveča. Po prejeti zahtevi za plačilo se preveri stanje elektronske gotovine na njegovem e-računu (eR). V primeru zadostnega stanja lahko nadaljuje s postopkom plačila.

Prejemnik plačila predstavlja trgovca, ki sprejme plačilo v obliki elektronske gotovine. Na svoji napravi NFC ima nameščeno aplikacijo za brezkontaktno plačevanje. Njegova glavna vloga je, da pošlje zahtevo za plačilo in željen znesek ter ob vsakem plačilu preveri ustreznost in veljavnost pre-

jete elektronske gotovine s pomočjo javnega ključa banke. Slednja se varno hrani na e-računu (eR) v njegovi napravi do izvedbe postopka za polog. Postopek za polog pridobljene elektronske gotovine se lahko izvede po uspešno zaključenem plačilu. Če ima na svojem e-računu (eR) prejeto elektronsko gotovino pridobljeno med različnimi plačili, jih lahko kadarkoli položi. Po končanem postopku za polog banka poveča stanje na njegovem transakcijskem računu (tR).

3.1.2 Delovanje sistema

Sistem za brezkontaktno plačevanje vključuje štiri postopke: odprtje e-računa (eR) z namestitvijo aplikacije, plačilo, dvig in polog elektronske gotovine. Predlagan koncept bančnega sistema z izdelanimi primeri uporabe, je zasnovan na osnovi ugotovljenih zahtev, ki smo jih predvideli za ustrezno testiranje izmenjave elektronske gotovine z mobilnimi napravami NFC.

3.1.2.1 Vzpostavitev sistema

Pred začetkom uporabe sistema za brezkontaktno plačevanje mora imeti banka vzpostavljen sistem, ki omogoča izvedbo postopkov za odprtje e-računa (eR), dvig in polog. Sama inicializacija zajema ustvarjanje para ključev RSA – javnega in zasebnega. Javni ključ RSA banka razdeli svojim komitentom med postopkom za odprtje e-računa, zasebnega pa obdrži le zase in ga varno hrani. S pomočjo zasebnega ključa banka med postopkom za dvig digitalno podpiše elektronsko gotovino. Javni ključ se uporablja za preverjanje digitalnega podpisa med postopki za dvig, plačilo in polog. Uspešno preverjen digitalni podpis zagotavlja veljavnost elektronske gotovine.

Bančni sistem mora imeti nastavljen tudi parameter n, ki plačniku pove kolikšno število slepih sporočil mora ustvariti in kolikšno število parov identitete mora vključiti v posamezno slepo sporočilo med postopkom za dvig elektronske gotovine. Vrednost parametra n se določi med inicializacijo bančnega sistema.

Pomembna je tudi določitev trajanja veljavnosti elektronske gotovine. Slednje bi bilo lahko definirano z dodatno informacijo v samem zapisu elektronske gotovine. V tem sistemu je omejitev določena s trajanjem veljavnosti ključev RSA. Če banka zamenja par ključev RSA z novimi, potem predhodno dvignjena elektronska gotovina, ki še ni bila položena, ni več veljavna, saj je preverjanje njenega digitalnega podpisa neuspešno. Banka lahko zamenja stara ključa RSA z novimi, na podlagi vnaprej določenega časovnega intervala (npr. eno leto).

Zaradi poenostavitve postopkov za dvig in plačilo, se v predlaganem sistemu uporablja le elektronsko gotovino z vrednostjo ena. Če plačnik izvede postopke za dvig v vrednosti deset, pridobi deset enot elektronske gotovine z vrednostjo ena.

3.1.2.2 Ustvarjanje e-računa (eR) in namestitev aplikacije

Ustvarjanje e-računa (eR) se izvede med banko in uporabnikom, ki ima pri njej že odprt transakcijski račun (tR) in že uporablja različne storitve. Uporabnik se identificira z osebnim dokumentom. Na njegovo mobilno napravo NFC se iz bančnega sistema prenese in namesti aplikacija za brezkontaktno plačevanje.

Postopek za aktivacijo e-računa (eR) v brezkontaktnem sistemu za plačevanje, vključuje naslednje aktivnosti (slika 3.2):

1. Aplikacija pošlje zahtevo ("openAccount") za ustvarjanje e-računa (eR).

2. Uslužbenec v bančni sistem vnese podatke o uporabniku (ime, priimek, poštni naslov, telefonska številka, osebna identifikacijska številka ali emšo, mednarodna številka transakcijskega računa (iban) in fotografija), ki se shranijo v podatkovno bazo.

3. Bančni sistem pošlje aplikaciji informacijo iz predhodno definiranih podatkov, namenjeno vključevanju identitete uporabnika v elektronsko gotovino. Predstavlja jo mednarodna številka transakcijskega računa ali iban, ki enolično določa uporabnika v bančnem sistemu. Če podatki

ne obstajajo, bančni sistem obvesti uslužbenca o napaki in postopek se zaključi. Nato je potrebno izvesti celoten postopek od začetka.

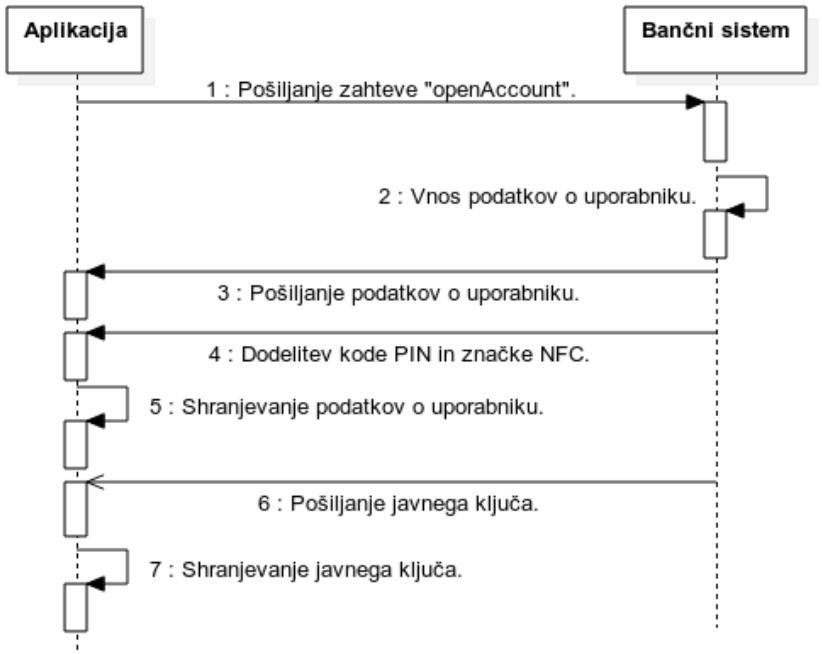
4. Uporabniku se dodelita koda PIN in značka NFC, potrebni za prijavo uporabnika v aplikacijo.

5. Aplikacija prejete podatke shrani na mobilno napravo.

6. Bančni sistem posreduje svoj javni ključ RSA.

7. Aplikacija shrani prejeti javni ključ na mobilni napravi.

Po uspešno zaključenem postopku za odprtje e-računa (eR) ima uporabnik na mobilni napravi nameščeno in aktivirano aplikacijo za brezkontaktno plačevanje.



Slika 3.2: Postopek ustvarjanja e-računa (eR) med bančnim sistemom in aplikacijo.

3.1.2.3 Dvig

Dvig je postopek, ki se izvede med bančnim sistemom in aplikacijo za brezkontaktno plačevanje nameščeno na uporabnikovi mobilni napravi NFC. Po zagonu in prijavi v aplikacijo je potrebno izbrati aktivnost Dvig in vpisati znesek:

1. Aplikacija pošlje zahtevo ("withdraw") za dvig in željen znesek dviga elektronske gotovine.
2. Bančni sistem preveri limit za dvig in stanje na uporabnikovem transakcijskem računu (tR). Če ima uporabnik zadostno stanje, se postopek nadaljuje, sicer se ustavi.
3. Bančni sistem obvesti aplikacijo o stanju uporabnikovega transakcijskega računa (tR).

Naslednji postopek za dvig ene enote elektronske gotovine se med bančnim sistemom in aplikacijo ponovi glede na izbran znesek (slika 3.3):

1. Aplikacija ustvari n slepih sporočil določenega formata, ki predstavljajo elektronsko gotovino in jih pošlje banki. Uporabi faktorje za skrivanje, ki jih pridobi iz javnega ključa banke. Format vključuje naključni unikatni niz oziroma identifikator, vrednost in seznam sestavljen iz n parov identitet. Vrednost parametra n je določena v postopku inicializacije sistema.
2. Aplikacija posreduje n slepih sporočil bančnemu sistemu.
3. Bančni sistem naključno izbere eno izmed n sporočil.
4. Bančni sistem obvesti aplikacijo o njegovi izbiri.
5. Aplikacija pošlje $n-1$ skrivnih faktorjev, ki jih je uporabila za ustvarjanje posameznih slepih sporočil.
6. Aplikacija pošlje $n-1$ izvirnih sporočil bančnem sistemu.

7. Bančni sistem z zasebnim ključem RSA digitalno podpiše $n-1$ slepih sporočil in jih razkrije na podlagi skritih faktorjev, ki jih je predhodno posredovala aplikacija.

8. Bančni sistem preveri veljavnost digitalnih podpisov slepih sporočil z javnim ključem RSA, na podlagi izvirnih sporočil, ki jih je predhodno posredovala aplikacija.

9. Če je preverjanje uspešno, bančni sistem preveri format in ustreznost izvirnih $n-1$ sporočil. Slednje zajema preverjanje naključnih unikatnih nizov oziroma identifikatorjev, vrednosti in prisotnosti prave identitete uporabnika (iban) v seznamu.

10. Če je preverjanje uspešno, potem bančni sistem digitalno podpiše izbrano slepo sporočilo z zasebnim ključem RSA.

11. Bančni sistem posreduje digitalno podpisano slepo sporočilo.

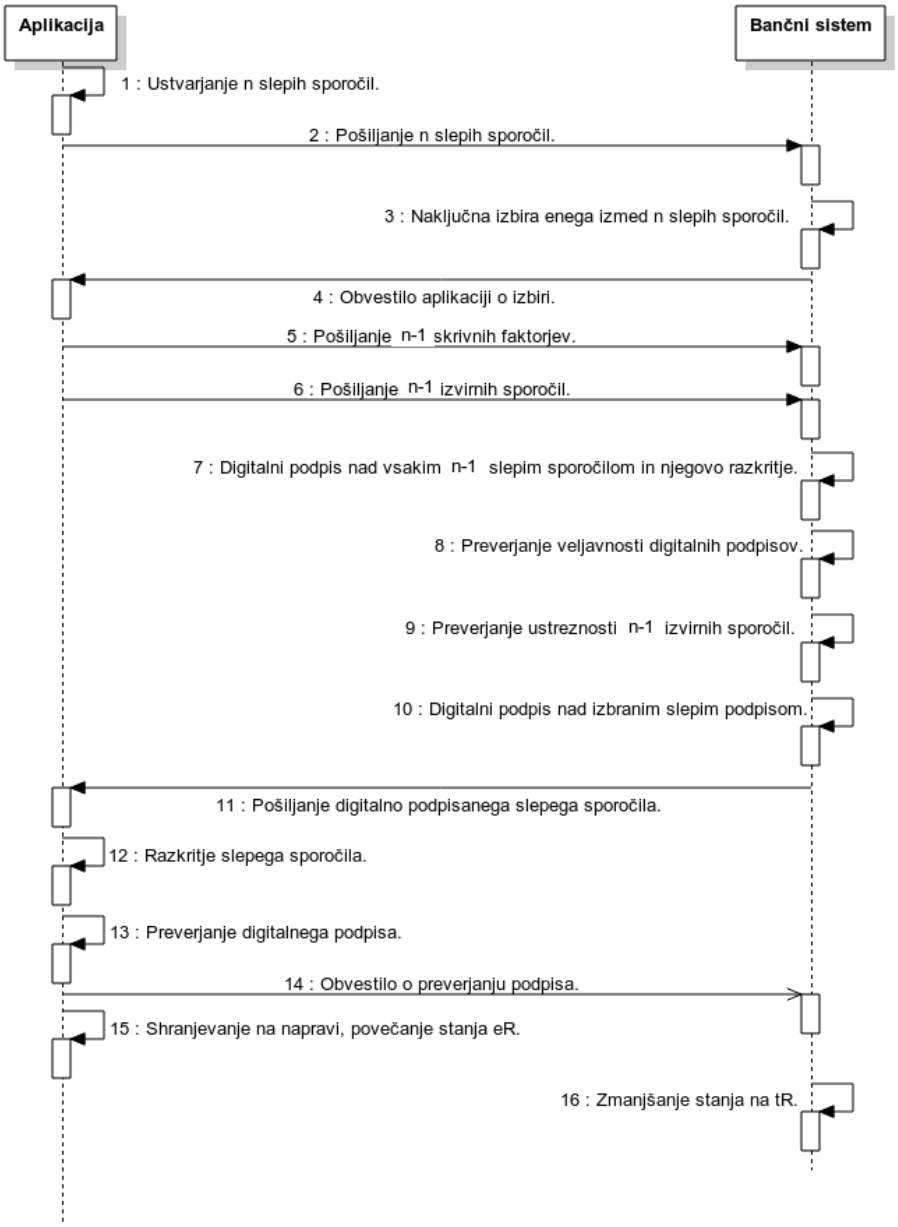
12. Aplikacija razkrije slepo podpisano sporočilo na podlagi skrivnega faktorja.

13. Aplikacija preveri veljavnost digitalnega podpisa z javnim ključem RSA sistema, na podlagi izvirnega sporočila.

14. Aplikacija obvesti bančni sistem o uspešnosti preverjanja digitalnega podpisa.

15. Če je preverjanje uspešno, aplikacija varno shrani razkrito sporočilo na mobilni napravi, ki predstavlja eno enoto elektronske gotovine – poveča stanje elektronske gotovine na njenem e-računu (eR). Sicer je postopek za dvig neuspešen in ga je potrebno ponoviti. Stanje tako na transakcijskem računu (tR) kot e-računu (eR) uporabnika ostane nespremenjeno.

16. Bančni sistem zmanjša stanje na transakcijskem računu (tR) uporabnika.



Slika 3.3: Postopek za dvig ene enote elektronske gotovine med bančnim sistemom in aplikacijo.

3.1.2.4 Plačilo

Brezkontaktno plačilo poteka med dvema uporabnikoma (plačnik – aplikacija A in prejemnik plačila – aplikacija B), ki imata na svojih mobilnih napravah NFC nameščeno aplikacijo. Postopek se brez udeležbe bančnega sistema izvede na naslednji način (slika 3.4):

1. Prejemnik plačila v aplikaciji B izbere možnost plačila in vnese željen znesek. Uporabnika združita napravi.

2. Preko tehnologije NFC plačnik v aplikaciji A prejeme zahtevo ("payToMe") za plačilo skupaj z zahtevanim zneskom in izzivi. Seznami dolžine n naključno izbranih števil 0 in 1, ki povedo kateri del identitete uporabnika mora aplikacija razkriti.

3. Plačnik v aplikaciji A lahko potrdi oziroma zavrne zahtevano plačilo.

4. Če potrdi plačilo, se na napravi oziroma e-računu (eR) v aplikaciji A, izvede preverjanje stanja elektronske gotovine.

5. Če je stanje elektronske gotovine na e-računu (eR) zadostno, aplikacija A pripravi podatke za pošiljanje (ustrezno število enot elektronske gotovine in odgovore na izzive), sicer se postopek zaključi.

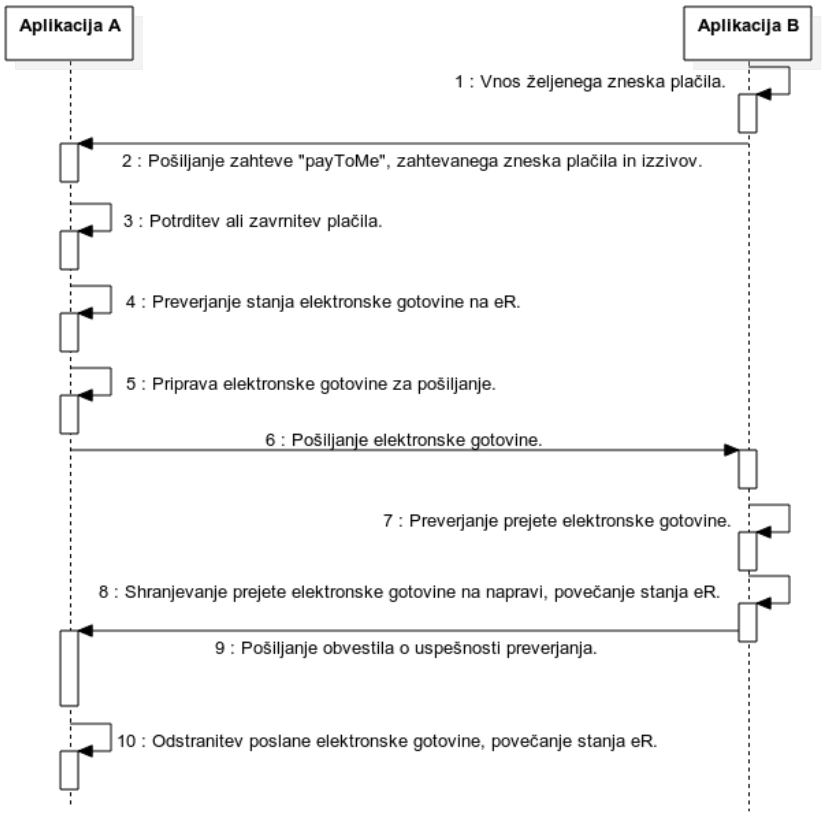
6. Pošiljanje pripravljene elektronske gotovine. Uporabnika ponovno združita napravi.

7. Aplikacija B prejemnika plačila preveri veljavnost prejete gotovine z javnim ključem banke in prisotnost odgovorov na izzive.

8. Če je preverjanje uspešno, aplikacija B shrani prejeto elektronsko gotovino. Poveča stanje e-računa (eR) in preide v glavni meni, kjer se izpiše obvestilo o uspešno prejetem plačilu.

9. Pošiljanje obvestila o uspešnosti preverjanja prejete elektronske gotovine. Uporabnika ponovno združita napravi.

10. Sprememba stanja elektronske gotovine v aplikaciji A. Stanje e-računa (eR) se zmanjša. Aplikacija preide v glavni meni, kjer se izpiše obvestilo o uspešno izvedenem plačilu.



Slika 3.4: Postopek za izvedbo brezkontaktnega plačila med aplikacijama.

3.1.2.5 Polog

Polog je postopek, ki se izvede med bančnim sistemom in aplikacijo za brezkontaktno plačevanje nameščeno na NFC mobilni napravi prejemnika plačila. Uporabnik izbere možnost za Polog (slika 3.5):

1. Aplikacija pošlje zahtevo ("deposit") za polog in vso prejeto elektronsko gotovino, ki je shranjena na napravi oziroma e-računu (eR).

Bančni sistem izvede preverjanje položene elektronske gotovine, ki ga sestavljata točki 2 in 3:

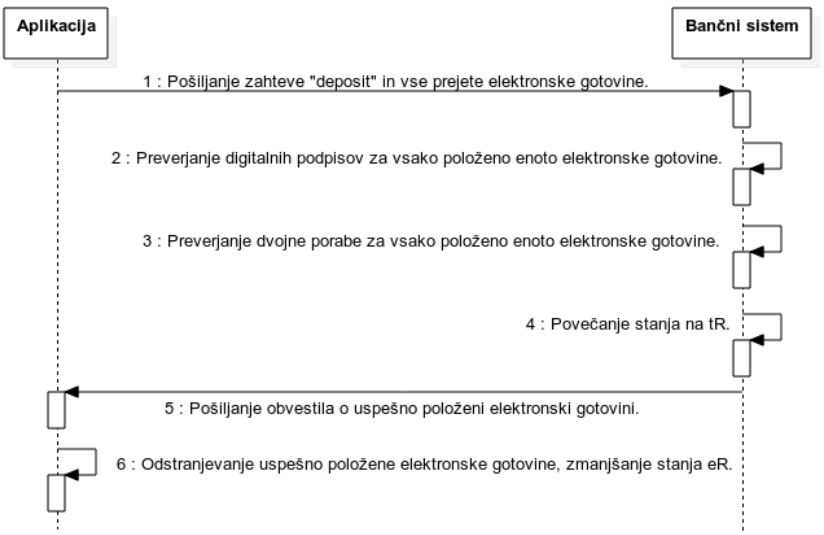
2. Preverjanje digitalnih podpisov posamezne elektronske gotovine s svojim javnim ključem RSA.

3. Preverjanje dvojne porabe s pomočjo naključnega unikatnega niza oziroma identifikatorja v podatkovni bazi, ki hrani vso že do sedaj položeno elektronsko gotovino.

4. Po končanem preverjanju bančni sistem spremeni stanje na uporabnikovem transakcijskem računu (tR), glede na uspešnost preverjanja.

5. Bančni sistem posreduje aplikaciji obvestilo, katere enote elektronske gotovine so bile uspešno položene.

6. Aplikacija odstrani uspešno položeno elektronsko gotovino oziroma spremeni stanje e-računa (eR).



Slika 3.5: Postopek za polog elektronske gotovine med bančnim sistemom in aplikacijo.

3.2 Arhitektura in zasnova sistema

Sistem za brezkontaktno plačevanje sestoji iz:

- strežniškega dela, ki vsebuje transakcijski račun (tR) in omogoča odjemalcu dvig in polog elektronske gotovine ter
- mobilnega dela odjemalca, ki vsebuje e-račun (eR) in omogoča shranjevanje ter izmenjavo elektronske gotovine.

Rešitev je zasnovana na predpostavki, da komunikacija poteka med bančnim sistemom in aplikacijo na mobilni napravi NFC na varen način z uporabo varnih kanalov [2]. Najbolj razširjena in priljubljena protokola za njihovo izvedbo sta protokola SSH (Secure Shell) [31] ali TLS (Transport Layer Security) [17], ki omogočata izvedbo varne komunikacije preko nezavarovanih omrežij, kot je internet. Komunikacija med odjemalcem in strežnikom poteka na naslednji način:

- Odjemalec pošlje zahtevo za uporabo varnega prenosa.
- Strežnik se predstavi z javnim ključem – šifriranje z javnim ključem in uporaba digitalnega certifikata.
- Odjemalec ustvari ključ za simetrično šifriranje, ga zašifrira s strežnikovim javnim ključem ter mu ga pošlje.
- Strežnik dekodira prejeti ključ s svojim zasebnim ključem.
- Nadaljnja komunikacija med odjemalcem in strežnikom poteka z uporabo simetričnega šifriranja s prejetim ključem.

3.2.1 Bančni sistem

Strežniška aplikacija deluje na strežniku in predstavlja simulirano in testiranju prilagojeno rešitev bančnega sistema, ki skrbi za izvedbo predlaganih postopkov za odprtje računa, dvig in polog. Podatki se hranijo v podatkovni bazi na strežniku. Komunikacija med mobilno napravo in bančnim

sistemom poteka na osnovi modela odjemalec–strežnik (angl. client/server). Bančni sistem uporablja par ključev RSA, javnega in zasebnega, ki sta varno shranjena v podatkovni bazi.

3.2.1.1 Funkcionalnosti

Strežniška aplikacija v posameznih fazah delovanja omogoča naslednje funkcionalnosti bančnega sistema:

- **Inicializacija**:
  - ustvarjanje para ključev RSA,
  - shramba para ključev RSA v podatkovni bazi,
  - preverjanje prisotnosti para ključev RSA v podatkovni bazi in
  - izbira vrednosti parametra n.
- **Ustvarjanje e-računa (eR)**:
  - vnos uporabnikovih podatkov in
  - hranjenje uporabnikovih podatkov v podatkovni bazi.
- **Dvig**:
  - preverjanje stanja na uporabnikovem transakcijskem računu (tR),
  - naključna izbira skritega (slepega) sporočila izmed n sporočil,
  - preverjanje formata in veljavnost ter ustreznost ustvarjenja elektronske gotovine (naključen unikaten niz oziroma identifikator, vrednost in prisotnost seznama identitete uporabnika),
  - izvedba digitalnega podpisa (zasebni ključ RSA) nad izbranim skritim (slepim) sporočilom in
  - sprememba stanja na uporabnikovem transakcijskem računu (tR).
- **Polog**:

  - preverjanje veljavnosti in ustreznosti posredovane elektronske gotovine (preverjanje digitalnega podpisa in preverjanje prisotnosti odgovora na izziv),
  - preverjanje dvojne porabe na podlagi naključnega unikatnega niza oziroma identifikatorja in podatkovne baze,
  - razkritje uporabnikove identitete v primeru dvojne porabe na osnovi posredovanih odgovorov na izziv in
  - sprememba stanja na uporabnikovem transakcijskem računu (tR).

3.2.1.2 Podatkovni model

Na sliki 3.6 je prikazan entitetno relacijski (ERD) diagram podatkovne baze strežniške aplikacije. Osnovne entitete podatkovne baze so:

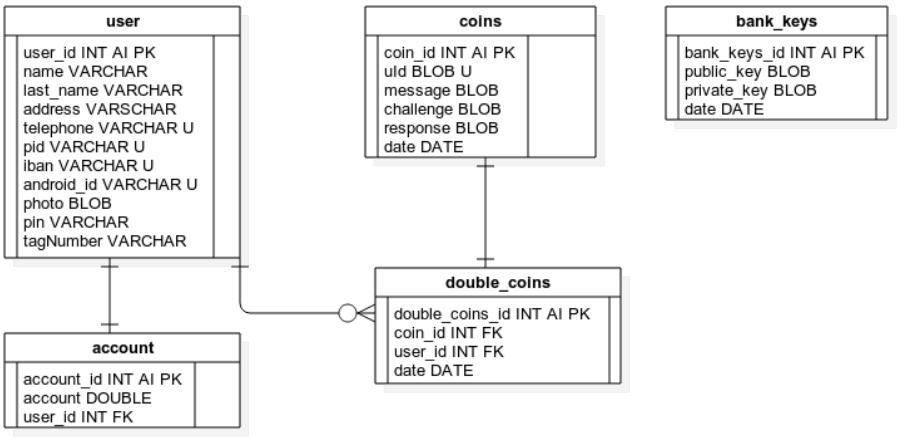
- **Uporabniki** (*user*) – seznam uporabnikov sistema:
  - unikaten identifikator v tabeli,
  - ime,
  - priimek,
  - naslov,
  - telefonska številka (9 bajtov),
  - enotna matična številka občana (13 bajtov),
  - mednarodna števila transakcijskega računa – iban (16 bajtov),
  - identifikacijska številka Android naprave,
  - fotografija,
  - številka PIN (4 bajte) in
  - številka značke NFC (16 bajtov).
- **Račun** (*account*) – stanje transakcijskega računa (tR) uporabnika:
  - unikaten identifikator v tabeli,

  - unikaten identifikator uporabnika in
  - stanje.

- **Elektronska gotovina** (*coins*) – prejeta elektronska gotovina v fazi pologa:
  - unikaten identifikator v tabeli,
  - naključen unikaten niz oziroma identifikator elektronske gotovine (digitalno podpisan),
  - sporočilo elektronske gotovine,
  - izziv ustvarjen med plačilom,
  - odgovor na izziv ustvarjen med plačilom in
  - datum pologa.

- **Dvojno porabljena elektronska gotovina** (*double_coins*) – zapisi o dvojni porabi prejete elektronske gotovine:
  - unikaten identifikator v tabeli,
  - naključen unikaten niz oziroma identifikator obstoječe elektronske gotovine,
  - datum izvedbe dvojne porabe.

- **Ključi bančnega sistema** (*bank_keys*) – javni in privatni ključ:
  - unikaten identifikator v tabeli,
  - javni ključ RSA,
  - privatni ključ RSA in
  - datum ustvarjanja ključev.



Slika 3.6: Podatkovni model (ERD) strežniškega dela.

3.2.2 Mobilna aplikacija

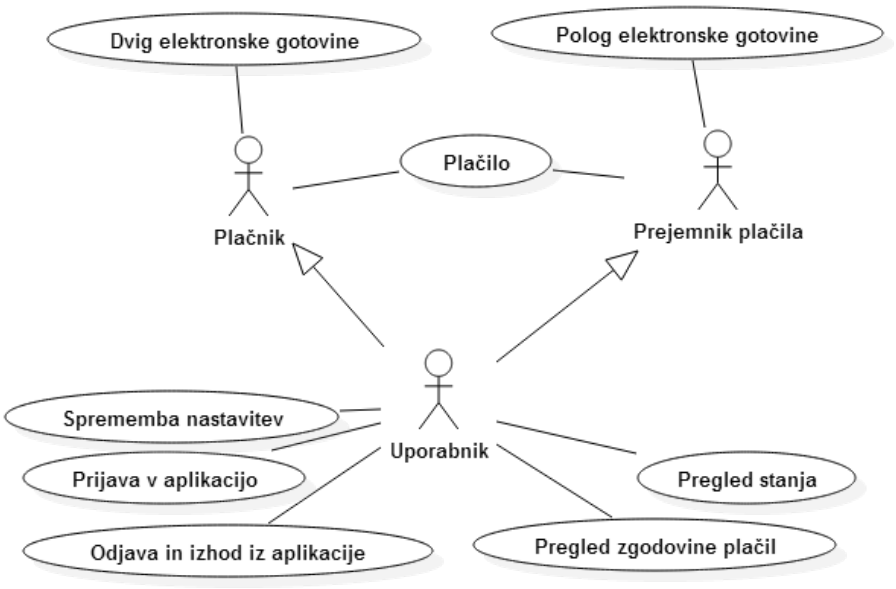
Najpomembnejši del sistema za brezkontaktno plačevanje je mobilna aplikacija, ki deluje na Android napravah z uporabo tehnologije NFC. Omogoča izvedbo postopkov za odprtje računa, dvig in polog v povezavi s strežniško aplikacijo ter postopek plačevanja. V fazi odprtja računa, dviga in pologa mora imeti mobilna aplikacija dostop do internetnega omrežja preko 2G, 3G, 4G ali preko brezžične povezave WiFI. Prenos podatkov v fazi plačila se izvede med aplikacijama na dveh mobilnih napravah NFC, s pomočjo funkcije Android Beam.

Podrobnejše delovanje aplikacije za brezkotnaktno plačevanje je predstavljeno na sliki 3.7. Diagram prikazuje uporabnika, ki je lahko plačnik ali prejemnik plačila. V brezkontaktnem načinu plačevanja ima na voljo osem funkcionalnosti.

3.2.2.1 Funkcionalnosti

Funkcionalnosti mobilne aplikacije, ki so uporabniku na voljo, so naslednje:

- prijava v aplikacijo s pomočjo številke PIN in značke NFC,
- odjava in izhod iz aplikacije – uporabnika se samodejno odjavi iz apli-



Slika 3.7: Primeri uporabe mobilne aplikacije.

kacije ob njenem zaprtju,

- plačilo – uporabnik (prejemnik plačila) lahko začne postopek za izvedbo brezkontaktnega plačila,
- dvig – uporabnik (plačnik) lahko začne postopek za izvedbo dviga elektronske gotovine,
- polog – uporabnik (prejemnik plačila) lahko začne postopek za izvedbo pologa prejete elektronske gotovine,
- pregled stanja – uporabnik lahko preveri stanje svojega e-računa (eR) – stanje dvignjene in prejete elektronske gotovine,
- pregled zgodovine plačil – uporabnik lahko preveri zgodovino prilivov (prejem plačila) in odlivov (izvedba plačila) e-računa (eR) in
- sprememba nastavitev – uporabnik lahko spremeni nastavitve za dostop do strežnika (naslov in vrata). To je namenjeno enostavnejšemu testiranju aplikacije.

3.2.2.2 Podatkovni model

Osnovne entitete podatkovne baze na mobilni napravi so:

- **Podatki o uporabniku** (*userData*) – podatki o uporabniku za vključitev v identiteto elektronske gotovine:
  - unikaten identifikator v tabeli in
  - mednarodna številka transakcijskega računa – iban.
- **Dvignjena elektronska gotovina** (*coin*) – prvi del e-računa (eR), ki hrani podatke o posamezni dvignjeni elektronski gotovini:
  - unikaten identifikator v tabeli,
  - vrednost,
  - digitalno podpisan unikaten identifikator elektronske gotovine in
  - sporočilo elektronske gotovine.
- **Prejeta elektronska gotovina** (*coinR*) – drugi del e-računa (eR), ki in hrani podatke o posamezni prejeti elektronski gotovini:
  - unikaten identifikator v tabeli,
  - vrednost,
  - digitalno podpisan unikaten identifikator elektronske gotovine,
  - sporočilo elektronske gotovine,
  - izziv,
  - odgovor na izziv in
  - časovna značka opravljenega plačila.
- **Fotografija** (*photo*) – fotografija uporabnika, ki je bila poslana med postopkom za odprtje računa:
  - unikaten identifikator v tabeli in

  - fotografija.

- **Zgodovina plačil** (*history*) – podatki o vseh plačilih:
  - unikaten identifikator v tabeli,
  - leto,
  - mesec,
  - dan in
  - identifikator, ki pove ali gre za plačilo ali prejem plačila.

3.2.2.3 Prijava in varno hranjenje podatkov

Uporabnik se v aplikacijo prijavi s pomočjo kode PIN in značke NFC, ki mu jih dodeli banka. Značka NFC vsebuje naključno vrednost, ustvarjeno s strani banke. Ob prvem zagonu aplikacije se s pomočjo podane kode PIN in vsebine značke NFC ustvari geslo, ki se uporabi za izpeljavo šifrirnega ključa. Če uporabnik pri naslednjih prijavah ne poda pravilne kombinacije kode PIN in vsebine značke NFC, potem izpeljava ustreznega šifrirnega ključa ni mogoča in tako nima dostopa do podatkov.

Elektronska gotovina je shranjena na e-računu (eR), ki se nahaja na mobilni napravi NFC. Ne sme biti dostopna drugim aplikacijam, ampak zgolj aplikaciji za brezkontaktno plačevanje. Zato je potrebna izvedba varnega hranjenja podatkov aplikacije, s pomočjo simetričnega šifriranja. Ključ uporabljen za šifriranje se ne hrani na mobilni napravi, ampak je izpeljan s pomočjo gesla, ki ga poda uporabnik ob prijavi v aplikacijo. Ob uspešni prijavi uporabnika v aplikacijo se izpelje ključ, ki se uporablja za šifriranje in dešifriranje. Slednji se začasno hrani, dokler se uporabnik ne odjavi. Ob ponovnem zagonu aplikacije je potrebna ponovna prijava uporabnika in postopek se ponovi. Preden so podatki shranjeni v podatkovno bazo, so ustrezno šifrirani. Dešifriranje podatkov poteka pred njihovo uporabo.

Poglavje 4

Implementacija

Sistem za brezkontaktno plačevanje je razdeljen na bančni sistem, ki predstavlja strežniško aplikacijo in mobilno aplikacijo, ki predstavlja odjemalca. Predstavljena je izvedba komunikacije med njima in implementacija glavnih funkcionalnosti v vseh fazah sistema. Opisan je format podatkov elektronske gotovine in razredi ter metode za učinkovito delo s podatkovnima bazama tako v bančnem sitemu, kot v mobilni aplikaciji.

4.1 Bančni sistem

Bančni sistem predstavlja strežniško aplikacijo in omogoča mobilnim aplikacijam izvedbo postopkov za ustvarjanje e-računa (eR) ter dvig in polog elektronske gotovine. Za transakcijske račune (tR) uporabnikov skrbi tako, da upravlja s stanjem finančnih sredstev na le-teh. Med postopkom za dvig izvaja preverjanje ustreznosti ustvarjenih slepih sporočil in izvedbo digitalnega podpisa. V postopku za polog izvede preverjanje veljavnosti elektronske gotovine in dvojne porabe.

4.1.1 Tehnologije

Java je objektno in razredno orientiran programski jezik, razvit leta 1991 s strani podjetja Sun Microsystems [37]. Prva uradna različica (1.0) je izšla

leta 1996, zadnja (8.0) pa leta 2014. Trenutno za njeno posodabljanje in nadgradnjo skrbi podjetje Oracle. Na voljo so naslednje različice:

- standardna J2SE namenjena osebnim računalnikom,
- J2ME namenjena drugim napravam (mobilne naprave, pametni televizorji in ure itd.) in
- J2EE namenjena razvoju programske opreme velikih podjetij in organizacij.

Pomembna lastnost je prenosljivost, neodvisnost od strojne opreme in platforme. Mogoče jo je uporabljati na različnih operacijskih sistemih (Linux, Windows in Max OS X). Za uspešno prenosljivostjo in neodvisnostjo stoji njena arhitektura. Javanska programska koda se namesto v strojno, prevede v bajtno kodo, ki se izvaja na virtualnem stroju JVM (Java Virtual Machine). Končni uporabniki lahko uporabljajo samostojne javanske aplikacije s pomočjo tolmača JRE (Java Runtime Environment), nameščenim na njihovih napravah, ali kot spletne programčke (angl. applet), kjer je tolmač vgrajen v spletni brskalnik.

NetBeans IDE je odprtokodno (angl. open source) programsko orodje, razvito leta 1996 v programskem jeziku Java [40]. Leta 2010 ga je prevzelo podjetje Sun, od leta 2014 naprej pa zanj skrbi podjetje Oracle. Vključuje platformo in okolje IDE (Integrated Development Environment) za razvoj različnih programskih rešitev. Osnovni programski jezik je Java, vendar omogoča uporabo tudi drugih jezikov, kot so PHP, C, C++, JavaScript, HTML itd. Orodje je neodvisno od platforme, saj ga je mogoče uporabljati na vseh sistemih kjer deluje javanski virtualni stroj JVM (Linux, Windows, Mac OS X).

Bouncy Castle je odprtokodna knjižnica, sestavljena iz različnih kriptografskih API-jev [38]. Prvotno je vsebovala le API-je implementirane v programskem jeziku Java, kasneje pa tudi v C#. Razvita je bila leta 2000 z

namenom poenostavitve dela z različnimi kriptografskimi mehanizmi in konstrukti. Sestavljena je iz dveh delov:

- osnovnega API-ja, ki je sestavljen iz množice vseh osnovnih kriptografskih primitivov in algoritmov ter
- ponudnika JCE, zgrajenim nad osnovnim API-jem, ki vsebuje dodatne kriptografske funkcionalnosti.

MySQL je odprtokodna relacijska podatkovna baza, implementirana v programskih jezikih C in C++ [39]. Razvita je bila leta 1994 s strani Michaela Wideniusa in Davida Axmarka, zdaj pa je v lasti podjetja Oracle. Za delo s podatki se uporablja strukturirani povpraševalni jezik SQL (Structured Query Language), ki vsebuje stavke za ustvarjanje, brisanje, posodabljanje podatkovne baze in poizvedovanje po njej. Poleg same baze vsebuje tudi sistem za njeno upravljanje. Deluje na osnovi modela odjemalec–strežnik. Za dostop do strežnika je na voljo veliko odjemalcev, zbirk ukazov in vmesnikov za različne programske jezike, kot so: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby itd. Najpogosteje se uporablja spletni vmesnik *phpMyAdmin*, ki temelji na programskem jeziku PHP. Omogoča pregled, upravljanje in urejanje podatkov in strukture podatkovnih baz s pomočjo spletnega brskalnika. MySQL je neodvisen od platforme in ga je mogoče uporabljati na različnih operacijskih sistemih (Linux, Windows, Mac OS X).

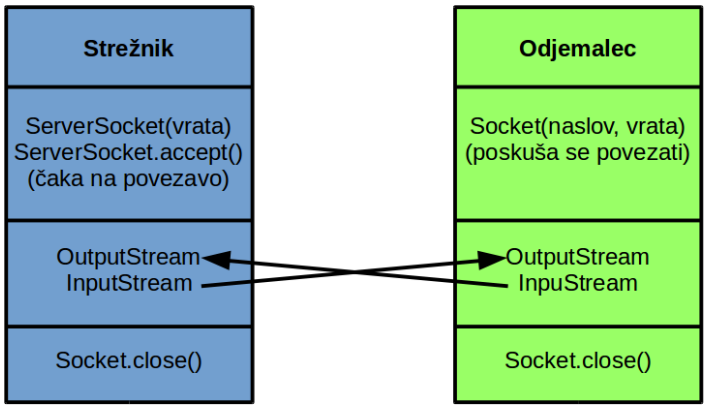
4.1.2 Komunikacija odjemalec–strežnik

Komunikacija med strežnikom in odjemalcem poteka na osnovi modela zahteva–odgovor (angl. request/response) (slika 4.1). Uporabljeni so vtičniki (angl. socket), ki omogočajo dvema napravama komunikacijo z uporabo standardnih mehanizmov [41]. Na voljo so tri vrste vtičnikov:

- brezpovezavni (angl. connectionless) vtičniki uporabljajo protokol UDP (User Datagram Protocol),

- povezavno-orientirani (angl. connection-oriented) vtičniki uporabljajo protokol TCP (Transmission Control Protocol) in
- osnovni (angl. raw) vtičniki, ki se uporabljajo v omrežnih napravah.

Izbrali smo povezavno-orientirane vtičnike v povezavi s TCP, ker omogočajo zanesljivo komunikacijo. Prav tako vtičniki omogočajo dvosmerno povezavo, zato lahko naprava sprejema ali pa pošilja podatke.



Slika 4.1: Shema komunikacije med strežnikom in odjemalcem s pomočjo vtičnikov.

Strežniška aplikacija se ob zagonu postavi v stanje poslušanja (angl. listening state) v katerem čaka na odjemalce, da se povežejo. Uporaba TCP omogoča sočasno streženje več odjemalcem hkrati. Za vsakega odjemalca strežnik ustvari pomožni proces, v katerem z njim vzpostavi povezavo TCP. Nato nadaljnji potek povezave med strežnikom in posameznim odjemalcem poteka ločeno. Čeprav se vsi odjemalci povežejo na isti naslov in številko vrat, strežnik za vsako povezavo posebej ustvari unikaten vtičnik. Sočasnost smo v Javi implementirali s pomočjo niti (angl. thread), kjer strežniška aplikacija za vsakega povezanega odjemalca ustvari svojo ločeno nit.
Odjemalec začne komunikacijo tako, da ustvari vtičnik s katerim se poveže s strežnikom. Ob uspešni povezavi strežnik na svoji strani tudi ustvari vtičnik.

Strežnik in odjemalec sta tako v stanju vzpostavljene povezave (angl. established state) in komunikacija med njima tako poteka s pisanjem in branjem podatkov iz ustvarjenih vtičnikov. V Javi je uporaba vtičnikov omogočena z dvema osnovnima razredoma. Prvi *java.net.Socket* predstavlja vtičnik, drugi *java.net.ServerSocket* pa strežniku zagotavlja mehanizme za vzpostavljanje povezave z odjemalci. Vzpostavljanje povezave med strežnikom in odjemalcem se izvede na naslednji način:

- Strežnik ustvari *ServerSocket* objekt in mu določi številko vrat (angl. port).
- Strežnik zažene metodo *accept* in čaka dokler se odjemalec ne poveže na strežnik (na zgoraj določena vrata).
- Odjemalec mora pred začetkom komunikacije ustvariti *Socket* objekt, ki mu poda ime strežnika (naslov) in številko vrat.
- Objekt *Socket* poskuša ustvariti povezavo na podan strežnik. Če je povezava vzpostavljena, lahko odjemalec komunicira s strežnikom preko objekta *Socket*.
- Metoda *accept* na strežniku po uspešno vzpostavljeni povezavi vrne referenco na nov vtičnik, s katerim je povezan odjemalčev vtičnik.

Komunikacija med strežnikom in odjemalcem poteka preko I/O podatkovnih tokov (angl. stream). Vtičnika na strežniku in odjemalcu imata na voljo dva razreda *InputStream* in *OutputStream*, ki predstavljata vhodni in izhodni tok. Objekt tipa *InputStream* na odjemalcu je povezan z objektom tipa *OutputStream* na strežniku in obratno.
Po uspešno vzpostavljeni povezavi poteka nadaljnja komunikacija glede na poslan zahtevek, ki ga pošlje odjemalec. Na voljo so trije zahtevki, ki omogočajo brezkontaktno plačevanje z ustrezno metodo v povezavi s strežnikom:

- openAccount – metoda **open** izvede transakcijo za odprtje računa,
- withdraw – metoda **withdraw** izvede transakcijo za dvig in

- deposit – metoda **deposit** izvede transakcijo za polog.

4.1.3 Delo s podatkovno bazo

Za povezavo bančnega sistema in relacijske podatkovne baze smo uporabili JDBC (Java Database Connectivity Technology) API, ki v programskemu jeziku Java definira dostop do podatkovne baze. Vključuje metode za poizvedovanje in posodabljanje podatkovne baze. Za povezovanje skrbi *DriverManager*, ki mu podamo ustrezen gonilnik. Ker smo se odločili za MySQL podatkovno bazo, smo uporabili gonilnik *com.mysql.jdbc.Driver*. Omogočena je uporaba stavkov za spreminjanje (CREATE, INSERT, UPDATE in DELETE) ter stavka za poizvedovanje (SELECT). Za delo z MySQL smo implementirali:

- Razred *ConnectionFactory* za vzpostavljanje povezave s podatkovno bazo. Vsebuje metodi:
  - *createConnection* – ustvari novo povezavo s podatkovno bazo in
  - *getConnection* – vrne instanco povezave s podatkovno bazo.
- Razred *BankDAO*, ki strežniški aplikaciji zagotavlja podatke za izvedbo postopkov. Vsebuje metode:
  - *changeAccountState* – spremeni uporabnikovo stanje na transakcijskem računu (tR),
  - *checkCoin* – preveri ali je bila enota elektronske gotovine s podanim identifikatorjem že položena,
  - *checkKeys* – preveri obstoj javnega in zasebnega ključa RSA,
  - *getAccountState* – vrne uporabnikovo stanje transakcijskega računa (tR),
  - *getCoin* – vrne podatke o določeni enoti elektronske gotovine,
  - *getKeys* – pridobi javni in privatni ključ RSA,

  - *storeCoin* – shrani položeno enoto elektronske gotovine,
  - *storeDoubleCoin* – shrani že položeno (dvojno porabljeno) enoto elektronske gotovine,
  - *storeKeys* – shrani javni in privatni ključ RSA in
  - *storeUserData* – shrani podatke o uporabniku, vnesene s strani uslužbenca med postopkom za ustvarjanje e-računa (eR).

- Razred *UtilsDB* za zapiranje različnih konstruktov in povezav. Vsebuje naslednje pomožne metode:
  - *closeConnection*,
  - *closePreparedStatement* in
  - *closeResultSet*.

4.1.4 Inicializacija

Ob zagonu bančnega sistema se najprej izvede postopek inicializacije, implementiran s strani metode *init*. Za uspešno uporabo sistema mora imeti strežniška aplikacija na voljo v podatkovni bazi javni in zasebni ključ RSA. Postopek se začne s preverjanjem prisotnosti obeh ključev z uporabo metode *checkKeys*. V primeru, da par ključev obstaja, ga je mogoče pridobiti s pomočjo metode *getKeys*, sicer ga je potrebno ustvariti. Slednje omogoča metoda *generateKeys* implementirana z uporabo knjižnice Bouncy Castle, kjer je uporabljen objekt *RSAKeyParamter* in generator *RSAKeyPairGenerator*. Velikost ključev, ki se uporabljajo v sistemu je določena na 2048 bitov. Po končanem ustvarjanju ključev, se ta shranita v podatkovno bazo z metodo *storeKey*, za nadaljnjo uporabo.

4.1.5 Dvig

Med postopkom za dvig elektronske gotovine, bančni sistem preveri vrednosti zahtevanega zneska na podlagi vnaprej določenega limita in stanja transakcijskega računa (tR) uporabnika. V primeru uspešnega preverjanja sledi

izvedba digitalnega podpisa nad naključno izbranim slepim sporočilom. Slednja ustvari mobilna aplikacija in jih posreduje bančnemu sistemu. Število potrebnih ustvarjenih sporočil je odvisnih od vnaprej, s strani bančnega sistema, določenega parametra n. V primeru napačnega števila sporočil se postopek prekine in potrebno ga je izvesti znova. Po izbranem slepem sporočilu, mobilna aplikacija posreduje bančnemu sistemu potrebne podatke za preverjanje vseh preostalih (neizbranih) slepih sporočil. Te podatke predstavljajo razkrita sporočila, skrivni faktorji in naključni nizi, uporabljeni pri ustvarjanju identitete. Bančni sistem izvede preverjanje za vsako slepo sporočilo posebej na naslednji način:

1. Digitalno podpiše slepo sporočilo z svojim zasebnim ključem RSA.

2. Razkrije slepo sporočilo na podlagi njegovega digitalnega podpisa, javnega ključa RSA in ustreznega skrivnega faktorja.

3. Izvede preverjanje digitalnega podpisa nad razkritim sporočilom in javnega ključa RSA.

V primeru uspešnega preverjanja sledi še pregled strukture in formata sporočil, ki zajema preverjanje prisotnosti identifikatorja v vseh sporočilih, ustrezne vrednosti in ustrezne identitete (iban številka) v prav vseh elementih seznama.

4.1.6 Polog

Glavni del postopka za polog predstavlja preverjanje elektronske gotovine, posredovane s strani prejemnika plačila. Izvede se v bančnem sistemu za vsako enoto elektronske gotovine posebej in je razdeljeno na:

- Preverjanje veljavnosti z javnim ključem (RSA) bančnega sistema in metode $verifiySignature$, implementirane z uporabo knjižnice Bouncy Castle.

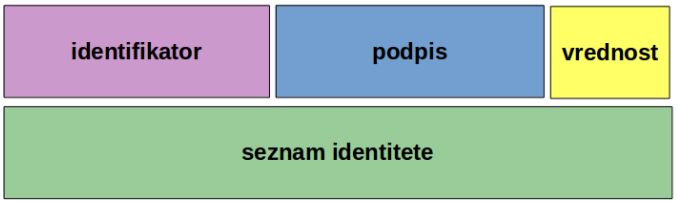
- Preverjanje dvojne porabe s pomočjo metode *checkCoin*, ki na podlagi identifikatorja preveri ali se mogoče v podatkovni bazi že nahaja enota elektronske gotovine z istim identifikatorjem.

V primeru neuspešnega preverjanja veljavnosti se enota elektronske gotovine zavrže, sicer se izvede še preverjanje dvojne porabe. Če je bila enota elektronske gotovine porabljena zgolj enkrat, jo bančni sistem shrani v podatkovno bazo z metodo *storeCoin* in poveča stanje transakcijskega računa (tR) prejemnika plačila. V obratnem primeru jo bančni sistem shrani med dvojno porabljeno elektronsko gotovino, s pomočjo metode *storeDoubleCoin* in začne postopek za razkrivanje identitete storilca. Če sta seznama odgovorov in izzivov v obeh enotah elektronske gotovine enaka, potem je dvojno porabo storil prejemnik plačila tako, da je dvakrat položil enako enoto. Sicer je storilec plačnik, ki je dvakrat plačal z isto enoto elektronske gotovine. Identiteta slednjega se ugotovi na podlagi izvedbe operacije XOR nad istoležnimi elementi obeh seznamov odgovorov. Na mestih, kjer sta se izziva ustvarjena med plačilom razlikovala, je plačnik posredoval oba dela, potrebna za ugotovitev razkritje identitete.

4.1.7 Format elektronske gotovine

Enota elektronske gotovine, pridobljena s postopkom za dvig in shranjena na mobilni napravi oziroma e-računu je sestavljena iz (slika 4.2):

- identifikatorja velikosti 256 bajtov,
- digitalnega podpisa na identifikatorjem velikosti 256 bajtov,
- vrednosti ena velikosti bajtov 1 bajta,
- seznama identitete, katerega velikost je odvisna od vrednosti parametra n (število elementov) in vključene informaciji o identiteti.



Slika 4.2: Format dvignjene elektronske gotovine.

4.2 Mobilna aplikacija

Mobilna aplikacija poimenovana *NFCePayment* za naprave Android ima vlogo odjemalca. Izvedba brezkontaktnega plačila je izvedena z uporabo tehnologije NFC. Za izvedbo postopkov za dvig in polog mora imeti naprava dostop do internetne povezave, saj se ta izvedeta v povezavi s strežniško aplikacijo. Dodatno aplikacija omogoča tudi pregled stanja elektronske gotovine, zgodovine plačil in prejemkov ter spreminjanje nastavitev.

Različica operacijskega sistema Android, na katerem lahko aplikacija deluje, je odvisna od uporabljenih funkcionalnosti iz razvojne knjižnice Android SDK. V našem primeru je minimalna možna podprta različica 17 (Android 4.2, 4.2.2), vendar je za optimalno delovanje najboljša uporaba različic od 21 (Android 5.0) naprej. Z zmanjševanjem minimalne različice, se povečuje število možnih Android naprav, ki jih lahko uporabimo, vendar se zmanjšuje število podprtih funkcionalnosti. Posledično se je potrebno izogibati funkcionalnostim, ki niso podprte v vseh različicah operacijskega sistema Android. V Android Studio sta minimalna in optimalna različica določeni v datoteki *build.gradle*.

4.2.1 Android Studio in SDK

Android Studio je IDE orodje za razvoj mobilnih aplikacij na Android platformi in uporablja programski jezik Java (različica J2ME) [33]. S strani podjetja Google je uradno priznano kot primarno okolje za razvoj in je tako zamenjalo predhodno okolje Eclipse Android Development Tools (ADT). Te-

melji na orodju Intellij IDEA, integriranem razvojnem okolju za potrebe Java razvijalcev programske opreme, ki ga je razvilo podjete jetBrains. Orodje je neodvisno od platforme in ga je mogoče uporabljati na vseh glavnih operacijskih sistemih (Linux, Windows, Mac OS X). Poleg osnovnih IDE funkcionalnosti vključuje še:

- Algoritme za organizacijo izvorne kode posamezne aplikacije.
- Možnost pametnega spreminjanja programske kode in uporabe hitrih popravkov, specifičnih za Android.
- Uporaba ProGuard-a in možnost podpisovanja aplikacij.
- Čarovnik na osnovi predlog s pogostimi Android dizajni in komponentami.
- Urejevalnik grafičnega vmesnika in postavitve komponent, ki omogoča hitro dodajanje novih komponent, predoglede in nastavitve za več različnih ekranov hkrati.

Android SDK (Android Software Development Kit) je razvojni paket, sestavljen iz množice različnih razvojnih pripomočkov, kjer so najpomembnejši [36]:

- razhroščevalnik (angl. debugger),
- knjižnice ali API-ji,
- emulator mobilnih naprav,
- vzorci programske kode in
- vodiči.

Uporaba Android SDK je možna na vseh glavnih platformah kot so Linux, Windows, Mac OS X ter tudi na sami Android platformi, vendar s pomočjo ustreznih Android aplikacij. Vključena je podpora starejšim različicam Android platforme, kar omogoča razvijalcem razvoj aplikacij za starejše naprave.

4.2.2 Funkcionalnosti Android naprave

Aplikacije Android lahko pri svojem delovanju uporabljajo različne tehnologije in funkcionalnosti naprave. Če želi aplikacija uporabljati določene funkcionalnosti aplikacije, ji je potrebno nastaviti določene pravice v datoteki $AndroidManifest.xml$. Uporaba slednje je obvezna in predstavlja osnovne informacije o sami aplikaciji, ki so nujno potrebne za njeno delovanje na platformi Android. Nastavljanje pravic aplikaciji je mogoče z uporabo značke $uses-permission$, katera eksplicitno določa katero funkcionalnost naprave lahko uporablja. Naša aplikacija uporablja tehnologijo NFC za izvedbo postopka brezkontaktnega plačila. Dovoljenje za njeno uporabo je nastavljeno z:

android.permission.NFC.

Poleg pravic NFC je potrebno nastaviti še pravice za dostop do internetne povezave, ki jih aplikacija potrebuje pri izvedbi postopkov za ustvarjanje e-računa (eR), dvig in polog:

android.permission.INTERNET,
android.permission.ACCESS_WIFI_STATE in
android.permission.ACCESS_NETWORK_STATE.

Uporaba zgornjih funkcionalnosti je ključnega pomena za delovanje mobilne aplikacije. Vsaka naprava, ki želi uporabljati aplikacijo mora imeti vgrajen modul NFC. Slednji mora biti pred izvedbo postopka za plačilo tudi vključen. Za lažje preverjanje modula NFC smo implementirali razred $NFCTools$, ki vsebuje metodi $nfcAdapterTest$ in $nfcAdapterEnabledTest$. Prva omogoča preverjanje prisotnosti mofula NFC na napravi, druga pa ali je njegova uporaba omogočena. Obe metodi izvedeta preverjanje na podlagi instance objekta $NfcAdapter$. Preverjanje prisotnosti modula NFC se preveri ob prvem zagonu aplikacije, preverjanje ali je modul omogočen pa pred samo izvedbo postopka za plačilo. Vzpostavljena internetna povezava je ključnega pomena pri izvedbi postopkov za ustvarjanje e-računa (eR) ter dviga in pologa elektronske gotovine. Preverjanje se izvede pred pričetkom

izvedbe posameznega postopka. V primeru postopka za odprtje e-računa (eR) se preverjanje izvede ob prvem zagonu aplikacije, v primeru dviga ali pologa pa, ko uporabnik v meniju izbere ustrezno možnost. Za lažjo izvedbo preverjanja internetne povezave smo implementirali razred $InternetTools$ in v njem pomožno metodo $isNetworkConnected$, ki za delo uporablja objekt $ConnectivityManager$.

4.2.3 Uporabniški vmesnik

Mobilna aplikacija sestoji iz sedmih aktivnosti, kjer ima vsaka s pomočjo datotek XML v mapi $res/layout$ definirano obliko, postavitev in komponente uporabniškega vmesnika. Aktivnost $MainActivity$ predstavlja glavno stran aplikacije, ki vsebuje meni. Ostale aktivnosti predstavljajo podstrani, ki so med seboj odvisne in hierarhično zasnovane. Prehajanje med njimi je enostavno, saj na novo podstran preidemo z dotikom na posamezni gumb, za vračanje na predhodno podstran pa lahko uporabimo standardni gumb nazaj. Uporabili smo osnovne komponente uporabniškega vmesnika kot so gumbi, vnosna polja, slike in tekst. Ker celotno upravljanje aplikacije poteka preko zaslona na dotik, so komponente dovolj velike in pregledno razporejene.
Ker imajo mobilne naprave Android zaslone različnih dimenzij in ločljivosti, je potrebno uporabniški vmesnik ustrezno prilagoditi. Velikosti nekaterih komponent uporabniškega vmesnika se sicer prilagodijo samodejno, za nekatere pa je to treba storiti ročno. Slednje se lahko stori z definiranjem različnih datotek XML in slik razporejenih v ustrezne mape glede na gostoto slikovnih pik (angl. pixel) zaslona:

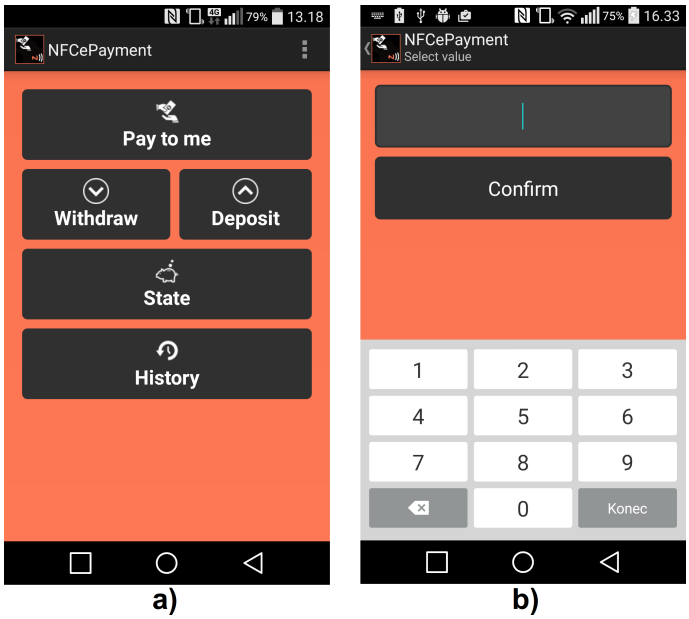
- nizka v mapi $res/drawable - ldpi$,
- srednja v mapi $res/drawable - mdpi$,
- visoka v mapi $res/drawable - high$,
- zelo visoka v mapi $res/drawable - xhdpi$ in
- zelo zelo visoka v mapi $res/drawable - xxhdpi$.

V našem primeru smo uporabniški vmesnik posebej prilagodili za normalno in visoko gostoto slikovnih pik zaslona. V ustrezni mapi smo postavili različne slike, ki se uporabljajo kot ikone gumbov na glavni strani. Prav tako smo vanju postavili dve različni velikosti ikone za zagon aplikacije.
Glavna stran vsebuje vrstico, ki se imenuje glava. Sestavljena je iz ikone in imena aplikacije ter gumba za dostop do nastavitev. Ostale podstrani prav tako vsebujejo glavo, ki pa sestoji iz ikone in imena aplikacije ter opisa posamezne podstrani in gumba nazaj.

Meni je glavna stran aplikacije, ki se uporabniku prikaže po zagonu aplikacije (slika 4.3 (a)). Sestavljena je iz petih gumbov, preko katerih lahko uporabnik dostopa do preostalih funkcionalnosti aplikacije in podstrani. Njena naloga je tudi preverjanje ali je modul NFC prisoten in omogočen. Ob kliku na gumba Dvig (Withdraw) ali Polog (Deposit) pa poskrbi za preverjanje internetne povezave. Ob prvem zagonu aplikacije je uporaba vseh funkcionalnosti onemogočena, dokler se postopek za odprtje računa ne zaključi.

Vnos zneska je podstran, ki se uporabniku prikaže ob kliku na gumb Plačaj mi (Pay to me) ali Dvig (Withdraw) (slika 4.3 (b)). Ob kliku na slednjega mora imeti naprava predhodno vzpostavljeno povezavo z internetom. Sicer se podstran ne prikaže, ampak sledi prikaz ustreznega obvestila o napaki. Vnos zneska sestoji iz vnosnega polja in gumba Potrdi (Confirm). Ob uporabnikovem dotiku vnosnega polja se samodejno prikaže številska tipkovnica, ki omogoča vnos zneska za plačilo ali dvig. Izbran znesek uporabnik potrdi s klikom na gumb Potrdi in izvajanje aplikacije se nadaljuje glede na predhodno izbrano možnost (plačilo ali dvig).

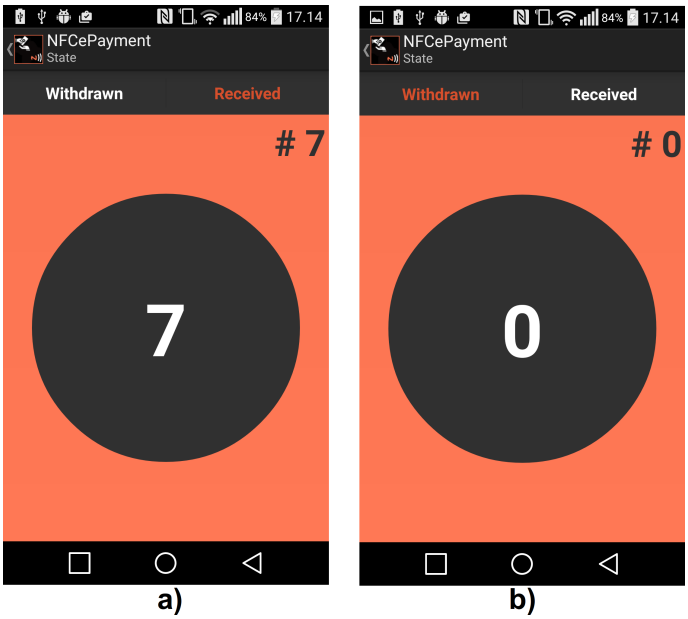
Pregled stanja je podstran, ki prikazuje podatke o stanju uporabnikove elektronske gotovine. Do nje lahko uporabnik dostopa le iz menija, s klikom na gumb Stanje (State). Podstran je sestavljena iz dveh ločenih pogledov, kjer eden prikazuje stanje elektronske gotovine pridobljene z različnimi dvigi (slika



Slika 4.3: Uporabniški vmesnik: a) meni, b) vnos zneska.

4.4 (a)), drugi pa stanje prejete elektronske gotovine (slika 4.4 (b)). Vsak pogled predstavlja del (angl. fragment) uporabniškega vmesnika aktivnosti za prikaz stanja, kar omogoča gradnjo dinamičnih uporabniških vmesnikov. Pogleda sta med seboj neodvisna in predstavljata ločena dela aktivnosti. Med pogledoma lahko uporabnik prehaja s pomočjo zavihkov (angl. tab) ali s pomočjo potekov (angl. swipe) preko zaslona v levo ali desno stran.

Pregled zgodovine je podstran, ki vsebuje podatke o vseh brezkontaktnih plačilih – prejemkih in odhodkih uporabnika (slika 4.5 (a)). Do nje lahko uporabnik dostopa samo iz strani meni, preko gumba Zgodovina (History). Ker lahko uporabnik v vsakem brezkontaktnem plačilu nastopa ali kot plačnik ali kot prejemnik plačila, je slednje ločeno z barvo. Vsako plačilo se prikaže kot zapis, ki je obrobljen z rdečo barvo, če gre za odhodek in z zeleno, če gre za prihodek. Za prikaz zapisov se uporablja pogled v obliki seznama (razred *ListView*) in je realiziran s pomočjo razreda *ListAdapter*. Ker je število zapisov lahko veliko in jih ni mogoče prikazati vseh naenkrat, se ti prikazujejo

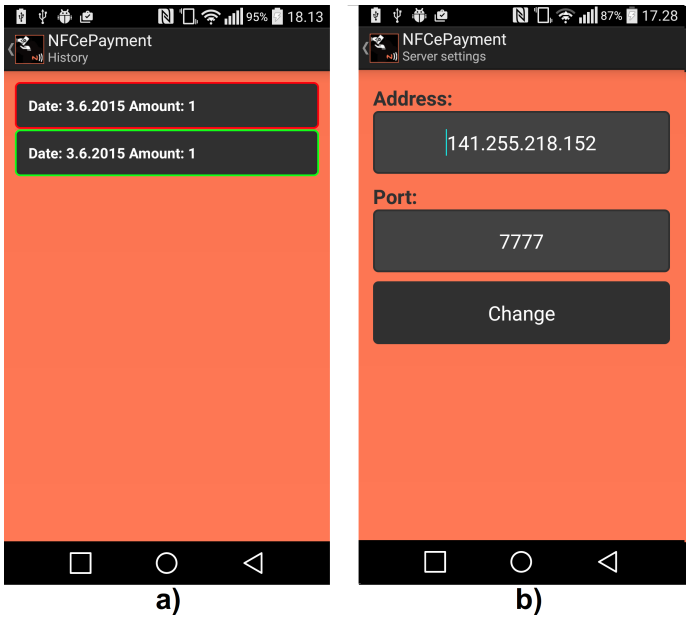


Slika 4.4: Uporabniški vmesnik: a) stanje dvignjene elektronske gotovine, b) stanje prejete elektronske gotovine.

dinamično. Med njimi se uporabnik premika s pomočjo potega s prstom gor ali dol po zaslonu.

Sprememba strežniških nastavitev je podstran namenjena testiranju aplikacije in omogoča uporabniku enostavno spreminjanje podatkov za dostop do strežnika (naslov in vrata) (slika 4.5 (b)). Sestavljena je iz dveh vnosnih polj, kjer je eno namenjeno naslovu in eno vratom strežnika ter gumba Spremeni (Change). Ob kliku na eno izmed vnosnih polj se samodejno prikaže tipkovnica, ki omogoča urejanje podatkov. Gumb Spremeni pa omogoča shranjevanje opravljenih sprememb. Do podstrani lahko uporabnik dostopa le iz menija, preko glave, ki na desni strani vsebuje gumb za dostop do nastavitev.

Upravitelj prenosa je podstran, ki se prikaže pred začetkom prenosa z uporabo tehnologije NFC (slika 4.6 (a)). Njena naloga je priprava zahteve

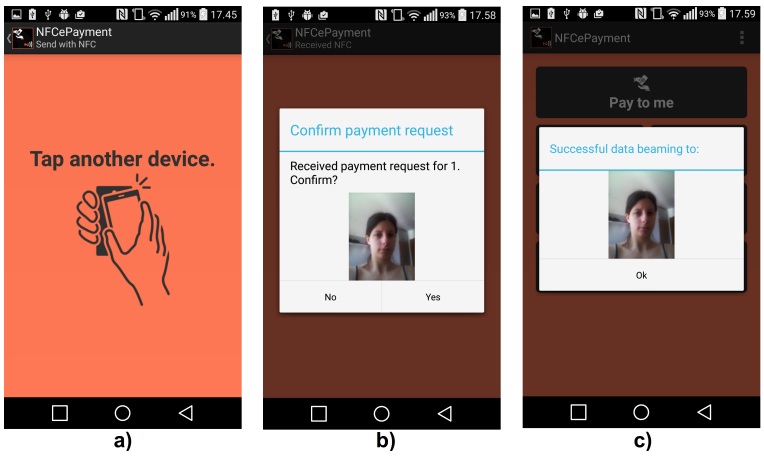


Slika 4.5: Uporabniški vmesnik: a) zgodovina brezkontaktnih plačil, b) nastavitve za dostop do strežnika.

ali elektronske gotovine za plačilo. Slednje je odvisno od tega ali uporabnik nastopa kot plačnik ali kot prejemnik plačila. Prejemnik plačila pripravi zahtevo za plačilo in na podstran preide preko podstrani za vnos zneska (slika 4.3 (b)). Plačnik pa pripravi elektronsko gotovino za plačilo in na stran preide preko podstrani za sprejem sporočila NFC (slika 4.6 (b)). Grafični vmesnik podstrani vsebuje navodilo uporabniku, naj za izvedbo prenosa NFC svojo napravo prisloni k drugi. Po uspešno opravljenem prenosu se podstran zapre in na glavni strani (meni) se prikaže pogovorno okno (angl. dialog box) s fotografijo osebe kateri, je bilo sporočilo poslano (slika 4.6 (c)).

Sprejem sporočila z uporabo NFC je podstran, ki se zažene samodejno, za kar poskrbi sistem za razporejanje značk (slika 4.6 (b)). Vsako sporočilo (zahteva ali plačilo) ima definiran tip značke, ki je specifičen za našo aplikacijo. Slednje pove sistemu za razporejanje značk, kateri aplikaciji je namenjeno in katero aktivnost naj zažene. Slednje je določeno v dato-

teki *AndroidManifest.xml* s pomočjo filtra namer. Uporabniški vmesnik podstrani je sestavljen iz pogovornega okna, ki prikazuje znesek zahtevanega plačila in fotografijo osebe, ki je poslala zahtevo za plačilo. Uporabniku je na voljo izbira, s katero lahko potrdi ali zavrne plačilo. V primeru potrditve plačila je naloga podstrani tudi preverjanje stanja elektronske gotovine. V primeru zadostnega stanja, aplikacija preide na podstran upravitelj prenosa (slika 4.6 (a)). Pošiljatelju zahteve za plačilo se prikaže tudi fotografija uporabnika, kateremu jo je poslal (slika 4.6 (c)).



Slika 4.6: Uporabniški vmesnik: a) upravitelj prenosa, b) sprejem sporočila, c) obvestilo o uspešnem prenosu.

4.2.4 Komunikacija s strežnikom

Pri izvajanju postopkov za ustvarjanje e-računa (eR), dvig in polog je potrebna komunikacija aplikacije s strežnikom. Na Androidu se tovrstna komunikacija izvaja v ozadju aplikacije. To je omogočeno z uporabo abstraktnega razreda *AsyncTask*, ki za vsako opravilo ustvari svojo nit, ločeno od glavne niti aplikacije. Med izvajanjem lahko uporabnik nemoteno uporablja aplikacijo naprej, tudi če se izvajanje niti še ni zaključilo. Ko ustvarjena nit zaključi opravilo, vrne rezultat nazaj na glavno nit aplikacije. Vsakemu ustvarjenemu

razredu *AsyncTask* določimo tip vhodnih in izhodnih podatkov ter implementiramo naslednji metodi:

- *doInBackground* in
- *onPostExecute*.

V metodi *doInBackground* poteka celotna komunikacija s strežnikom, kjer ustvarimo nov razred *Socket*, ki mu podamo naslov in številko vrat strežnika. Ustvariti je potrebno še instanci razreda *ObjectOutputStream* in *Object − InputStream* za nemoteno izmenjavo podatkov med strežnikom in aplikacijo. Celotna komunikacija tako poteka po naprej določenem postopku. Tako podatke primitivnih tipov, kot ostalih objektov je pred pošiljanjem potrebno pretvoriti v polje bajtov. Za lažje delo smo implementirali razred *SerializationTools*, ki vsebuje metodi *serialization* in *deserialization*. Slednji omogočata pretvorbo različnih objektov v polje bajtov in obratno.
Metoda *onPostExecute* je tista, ki vrača rezultat nazaj na glavno nit aplikacije. Vrednost rezultata pripravi metoda *doInBackground* in je odvisen od poteka njenega izvajanja. V našem primeru rezultat predstavljajo različna obvestila o uspešnem ali neuspešnem (napakah) izvajanju opravila v metodi *doInBackground*.

Za izvedbo postopkov za ustvarjanje e-računa (eR), dvig in polog smo implementirali razrede *OpenAccount*, *WithdrawProtocol* in *DepositProtocol*, ki razširjajo (angl. extends) abstraktni razred *AsyncTask* in imajo ustrezno povoženi (angl. override) metodi *doInBackground* in *onPostExecute*. Prva predstavlja dejansko implementacijo postopka, kjer si strežniška in mobilna aplikacija med seboj izmenjujeta ustrezne podatke. Druga pa je namenjena prikazu različnih obvestil o samem poteku postopka.

Postopek za ustvarjanje e-računa (eR) se izvede le enkrat, ob prvem zagonu aplikacije. Implementirali smo razred *App*, ki se izvede ob vsakem zagonu aplikacije, kar je določeno v datoteki *AndroidManifest.xml* v znački *application*. V njem se izvede preverjanje ali gre za prvi zagon aplikacije. Informacija o tem je določena v deljenih nastavitvah aplikacije, s pomočjo

spremenljivke *appFirstStart* tipa *boolean*. Ker vrednost spremenljivke ob prvem zagonu še ni določena, se uporabi privzeta vrednost $false$. V tem primeru se izvede preverjanje modula NFC in internetne povezave. Če je preverjanje uspešno, se zažene postopek za ustvarjanje e-računa (eR), ki je implementiran z razredom *OpenAccount*. Po uspešno zaključenem postopku se vrednost spremenljivke *appFirstStart* nastavi na *true*. Ob naslednjem zagonu aplikacije je preverjanje ali gre za prvi zagon aplikacije v razredu *App* neuspešno, postopek za ustvarjanje e-računa (eR) se izpusti in aplikacija se izvaja naprej.

Postopek za dvig elektronske gotovine smo implementirali s pomočjo razreda *WithdrawProtocol*, ki omogoča dvig ene enote elektronske gotovine. Ob vnosu večjega zneska dviga, se postopek ponovi glede na izbrani znesek. Pred izvedbo postopka je potrebna vzpostavljena povezava z internetom, za kar poskrbi ustrezno preverjanje. Ob vsakem dvigu ene enote elektronske gotovine, aplikacija ustvari naslednje sezname:

- Seznam sporočil elektronske gotovine, ki so sestavljena iz:
  - naključnega unikatnega niza (id) dolžine 16 bajtov,
  - vrednosti in
  - seznama identitete.
- Seznam slepih sporočil ustvarjenih s funkcijo *blindMessage* na osnovi naključnega unikatnega niza in skrivnega faktorja.
- Seznam skrivnih faktorjev za posamezno slepo sporočilo.

Dolžina vseh seznamov je enaka parametru n, ki je določen v strežniški aplikaciji in se posreduje aplikaciji med samim postopkom. Seznam identitete, ki je vključen v vsako sporočilo elektronske gotovine, se ustvari s funkcijo *createIdentityList*. Slednja ustvari prav tako n elementov, na podlagi identitete (iban), naključnega niza iste dolžine kot je identiteta, operacije XOR, zgoščevalnih vrednosti in njihovega združevanja. Poleg seznama identitet

aplikacija hrani uporabljene naključne nize, ki jih potrebuje strežniška aplikacija za preverjanje seznama identitete.
Za razkritje slepih sporočil je uporabljena metoda $unblindMessage$, ki tako kot metoda $blindMessage$ uporablja objekt $RSABlindingEngine$. Ustvarjanje skrivnih faktorjev je izvedeno na podlagi javnega ključa bančnega sistema, z metodo $generateBlindingFactor$, ki uporablja generator $blindingFactor - Generator$. Uporaba vseh kriptografskih operacij je implementirana z uporabo knjižnice Bouncy Castle. Po končanem postopku za dvig elektronske gotovine mora aplikacija shraniti elektronsko gotovino na svoji napravi oziroma e-računu (eR).
Po vnosu in potrditvi zneska dviga aplikacija preveri njegovo veljavnost preden se ta pošlje strežniški aplikaciji. Znesek mora biti pozitiven in ne enak nič. Če je preverjanje neuspešno se uporabniku prikaže obvestilo in vnos zneska je potrebo ponoviti.

Postopek za polog smo implementirali s pomočjo razreda $DepositProtocol$ in zajema pošiljanje vse elektronske gotovine, ki je bila pridobljena z različnimi plačili. Glavni del postopka se izvede v strežniški aplikaciji in zajema preverjanje veljavnosti elektronske gotovine in dvojne porabe. Po končanem preverjanju strežniška aplikacija posreduje seznam uspešno položene elektronske gotovine, ki jo nato mobilna aplikacija odstrani iz svoje naprave oziroma e-računa (eR).

4.2.5 Izvedba brezkontaktnega plačila

Izvedba plačila med uporabnikovima napravama (plačnik in prejemnik plačila) je realizirana s pomočjo tehnologije NFC. Za prenos sporočil se uporablja funkcija Android Beam. Postopek brezkontaktnega plačila sestavljajo trije prenosi NFC, ki se izvedejo, ko uporabnika združita napravi. Določili smo naslednje tri vrste prenosov:

- zahtevek za plačilo,
- plačilo in

- potrditev plačila.

Pošiljanje smo implementirali s pomočjo aktivnosti $SendWithNfcActivity$, ki glede na vrsto pripravi ustrezno sporočilo NDEF z uporabo metode $create-NdefMessage$. Uporablja se metoda $setNdefPushMessageCallback$, ki omogoča ustvarjanje sporočil NDEF v odvisnosti od trenutnega konteksta aplikacije. Vsakemu sporočilu je potrebno določiti tip, ki aplikaciji pove za katero vrsto prenosa gre in kako ga mora obdelati:

$application/com.nfcepayment.request$,

$application/com.nfcepayment.payment$ ali

$application/com.nfcepayment.confirm$.

Aktivnost za prejem in obdelavo sporočil NDEF smo aplikaciji določili z uporabo ustreznih filtrov namer v datoteki $AndroidManifest.xml$, ki jim podamo akcijo, kategorijo in tip podatkov.
Implementacija poteka brezkontaktnega plačila je naslednja (A – aplikacija prejemnika plačila, B – aplikacija plačnika):

- Prenos zahtevka za plačilo:

  A) Ustvari sporočilo NDEF sestavljeno iz želenega zneska, izzivov (tolikšno število seznamov dolžine n naključnih vrednosti 0 in 1, kolikor je vrednost zneska) in fotografije.

  B) Ustvari privzeto sporočilo NDEF sestavljeno iz fotografije.

A, B) Prenos sporočila s pomočjo funkcije Android Beam.

  B) Prejem sporočila NDEF in njegova obdelava. Na podlagi zneska in fotografije pošiljatelja se prikaže pogovorno okno $AlertDialog$ za potrditev ali zavrnitev sporočila. Če uporabnik potrdi zahtevo za plačilo, se postopek nadaljuje in sledi prenos plačila.

  A) Sprejem sporočila NDEF in njegova obdelava. Prikaz obvestila – pogovornega okna sestavljenega iz fotografije pošiljatelja.

- Prenos plačila:

  B) Preverjanje stanja dvignjene elektronske gotovine. Če je preverjanje uspešno, sledi pripravljanje odgovorov na izziv za vsako enoto elektronske gotovine posebej.

  B) Ustvarjanje sporočil NDEF sestavljenega iz posameznih enot elektronske gotovine, odgovorov za ustrezne izzive in fotografije.

  B) Ustvari privzeto sporočilo NDEF sestavljeno iz fotografije.

A, B) Prenos sporočila s pomočjo funkcije Android Beam.

  A) Prejem sporočila NDEF in njegova obdelava. Prikaže se fotografija pošiljatelja, sledi preverjanje veljavnosti posamezne prejete enote elektronske gotovine z javnim ključem RSA in prisotnosti odgovora na ustrezen izziv. Če je preverjanje vseh enot prejete elektronske gotovine uspešno, se postopek nadaljuje in sledi prenos potrditve.

  B) Sprejem sporočila NDEF in njegova obdelava. Prikaz obvestila – pogovornega okna sestavljenega iz fotografije pošiljatelja.

- Potrditev plačila:

  A) Shrani prejete enote elektronske gotovine v podatkovno bazo na svoji napravi. Ustvari sporočilo NDEF sestavljeno iz potrditve plačila in fotografije.

  B) Ustvari privzeto sporočilo NDEF sestavljeno iz fotografije.

A, B) Prenos sporočila s pomočjo funkcije Android Beam.

  B) Sprejem sporočila NDEF in njegova obdelava. Odstranitev predhodno posredovanih enot elektronske gotovine iz podatkovne baze.

  A) Sprejem sporočila NDEF in njegova obdelava. Prikaz obvestila – pogovornega okna sestavljenega iz fotografije pošiljatelja.

4.2.6 Delo s podatkovno bazo

Za delo s podatkovno bazo SQLite na mobilni napravi, smo implementirali razred *DBHelper* z naslednjimi metodami:

- *countCoins* – vrne število enot dvignjene elektronske gotovine,
- *countCoinR* – vrne število enot elektronske gotovine prejete z različnimi plačili,
- *deleteCoin* – izbriše posamezno dvignjeno enoto elektronske gotovine,
- *deleteCoinR* – izbriše posamezno pridobljeno enoto elektronske gotovine,
- *getAllCoinsForDeposit* – pridobi vse prejete enote elektronske gotovine,
- *getAllHistory* – pridobi informacije o vseh plačilih in prejemkih,
- *getCoins* – pridobi določeno število enot elektronske gotovine iz e-računa (eR),
- *getPhoto* – pridobi fotografijo uporabnika,
- *getUserData* – pridobi informacijo, ki je potrebna za ustvarjanje seznama identitete,
- *storeCoin* – shrani dvignjeno enoto elektronske gotovine,
- *storeCoinR* – shrani prejeto enoto elektronske gotovine in
- *storePhoto* – shrani fotografijo uporabnika.

Poglavje 5

Testiranje in analiza

Sistem za brezkontaktno plačevanje smo testirali za vse možne primere uporabe in različne mobilne naprave. Predstavljeni so postopki za uspešno in neuspešno izvedbo izbranih aktivnosti, z ustreznimi opozorili ali prekinitvami operacij. Povezovanje mobilnih naprav NFC in strežnikom poteka z uporabo brezžičnega omrežja.

5.1 Primeri uporabe

Testiranje je bilo zasnovano za izvedbo delovanja strežniške in mobilne aplikacije. Aktivnosti so bile definirane za dvig, polog in brezkontaktno plačilo NFC. Opisi se zaradi enostavnejše sledljivosti razlikujejo za:

SA – strežniška aplikacija (tekst na beli podlagi)

MA – mobilna aplikacija (tekst na sivi podlagi)

Ob zagonu aplikacije se najprej izvede preverjanje tehnologije NFC in nato se prikaže glavna stran, ki vsebuje meni. Uporabnik izbere posamezno aktivnost s klikom na ustrezen gumb.

MA (zagon aplikacije, preverjanje tehnologije NFC in izbira aktivnosti)

(1) Testiranje NFC (Ali ima naprava vgrajen modul NFC?):

Ne. (2.1)

Da. (2.2)

(2.1) Izhod iz aplikacije in obvestilo uporabniku.

(2.2) Izbira aktivnosti z dotikom na gumb:

Dvig. (3.1)

Plačilo (prejemnik plačila). (3.2)

Polog. (3.3)

Dvig elektronske gotovine se izvede z mobilno aplikacijo (MA), katera se preko brezžičnega omrežja poveže s strežniško aplikacijo (SA).

MA (preverjanje vzpostavljene internetne povezave, vnos zneska in preverjanje njegove veljavnosti)

(3.1) Testiranje internetne povezave (Ali ima naprava vzpostavljeno povezavo s internetom?):

Ne. (4.1)

Da. (4.2)

(4.1) Obvestilo uporabniku (internetna povezava ni vzpostavljena).

(4.2) Vnos želenega zneska za dvig.

(5) Testiranje zneska (Ali je vnesen znesek pozitiven?):

Ne. (6.1)

Da. (6.2)

(6.1) Obvestilo uporabniku (vnesen znesek ni veljaven).

(6.2) Obvestilo o začetku postopka za dvig.

(7) Pošiljanje zahteve (request: withdraw) in zneska za dvig.

SA (preverjanje limita)

(8) Testiranje vnesenega zneska (Ali je znesek manjši ali enak limitu?):

Ne. (9.1)

Da. (9.2)

MA (obvestilo)

(9.1) Obvestilo uporabniku (prevelik znesek).

SA (preverjanje stanja na uporabnikovem transakcijskem računu (tR))

(9.2) Testiranje vnesenega zneska (Ali je stanje na transakcijskem računu (tR) zadostno?):

Ne. (10.1)

Da. (10.2)

MA (obvestilo)

(10.1) Obvestilo uporabniku (ni dovolj sredstev na transakcijskem računu (tR)).

Nadaljevanje postopka za dvig se ponovi glede na vnesen znesek:

MA (ustvarjanje in pošiljanje slepih sporočil)

(10.2) Ustvari n enot elektronske gotovine (naključen unikaten identifikator, vrednost, seznam identitete).

(11) Skrije (blind) n enot elektronske gotovine.

(12) Pošiljanje n slepih sporočil.

SA (naključna izbira slepega sporočila)

(13) Naključna izbira enega izmed n slepih sporočil.

MA (pošiljanje preostalih izvirnih sporočil in pripadajočih skrivnih faktorjev)

(14) Pošiljanje preostalih $n - 1$ izvirnih sporočil in skrivnih faktorjev.

SA (testiranje preostalih slepih sporočil)

(15) Digitalni podpis nad vsakim izmed preostalih slepih sporočil.

(16) Razkritje vsakega digitalno podpisanega slepega sporočila s pomočjo pripadajočega skrivnega faktorja.

(17) Testiranje digitalnih podpisov razkritih sporočil na podlagi izvirnih sporočil (Ali je digitalni podpis veljaven?):

Ne. (18.1)

Da. (18.2)

(18.2) Testiranje vseh izvirnih $n - 1$ izvirnih sporočil (Ali je format ustrezen?):

Ne. (19.1)

Da. (19.2)

MA (obvestilo)

(18.1)(19.1) Obvestilo uporabniku (neuspešen postopek za dvig).

SA (izvedba in pošiljanje digitalnega podpisa)

(19.2) Digitalni podpis nad izbranim slepim sporočilom.

(20) Pošiljanje digitalno podpisanega slepega sporočila.

MA (razkritje digitalno podpisanega sporočila in preverjanje veljavnosti podpisa)

(21) Razkritje digitalno podpisanega slepega sporočila.

(22) Testiranje digitalnega podpisa (Ali je digitalni podpis veljaven?):

Ne. (23.1)

Da. (23.2)

(23.1) Pošiljanje obvestila strežniški aplikaciji (zavrnitev).

(23.2) Pošiljanje obvestila strežniški aplikaciji (potrditev).

SA (sprememba stanja na transakcijskem računu (tR))

(23.1)(23.2) Sprejem obvestila:

Zavrnitev. (24.1)

Potrditev. (24.2)

(24.1) Prekinitev postopka za dvig.

(24.2) Sprememba (zmanjšanje) stanja na transakcijskem računu (tR).

MA (sprememba stanja na e-računu (eR) in zaključek postopka za dvig)

(25) Sprememba (povečanje) stanja e-računa (eR) na napravi.

(26) Obvestilo uporabniku (postopek za dvig zaključen).

Brezkontaktno plačilo se izvede z mobilnima aplikacijama plačnika in prejemnika plačila z uporabo tehnologije NFC.
MA (T) – mobilna aplikacija prejemnika plačila (T – trgovec)
MA (K) – mobilna aplikacija plačnika (K – kupec)
MA (T – K) - pošiljanje iz mobilne aplikacije prejemnika plačila na plačnikovo
MA (K – T) - pošiljanje iz mobilne aplikacije plačnika na aplikacijo prejemnika plačnika.

MA (T) (vnos zneska in preverjanje njegove veljavnosti ter priprava in pošiljanje zahtevka za plačilo)

(3.2) Vnos želenega zneska za brezkontaktno plačilo.

(5) Testiranje zneska (Ali je vnesen znesek pozitiven?):

Ne. (6.1)

Da. (6.2)

(6.1) Obvestilo uporabniku (vnesen znesek ni veljaven).

(6.2) Priprava zahtevka za plačilo sestavljenega iz zneska in izzivi (naključni seznami sestavljeni iz 0 in 1).

MA (T – K) (pošiljanje zahtevka za plačilo)

(7) Uporabnika združita napravi in se dotakneta zaslona.

MA (K) (sprejem zahtevka za plačilo, preverjanje stanja elektronske gotovine in priprava plačila)

(8) Sprejem zahtevka za plačilo (Ali sprejme plačilo?):

Zavrne. (9.1)

Sprejme. (9.2)

(9.1) Obvestilo uporabniku (plačilo zavrnjeno) in zaključek postopka.

(9.2) Testiranje stanja elektronske gotovine (Ali ima uporabnik dovolj enot elektronske gotovine na e-računu (eR)?):

Ne. (10.1)

Da. (10.2)

(10.1) Obvestilo uporabniku (nezadostno stanje na e-računu (eR)) in zaključek postopka za plačilo.

(10.2) Pripravljanje plačila sestavljenega iz tolikšnega števila enot elektronske gotovine, kolikšen je bil zahtevan znesek. Vsaki enoti se ustvari odgovor glede na pripadajoči izziv.

MA (K – T) (pošiljanje elektronske gotovine)

(11) Uporabnika združita napravi in se dotakneta zaslona.

MA (T) (testiranje veljavnosti prejete elektronske gotovine s pomočjo javnega ključa banke, preverjanje odgovorov na izzive in sprememba stanja e-računa (eR))

(12) Testiranje vseh prejetih enot elektronske gotovine (Ali je digitalni podpis posamezne enote elektronke gotovine veljaven)?

Ne (13.1)

Da (13.2)

(13.2) Testiranje odgovora na izziv za vsako prejeto enoto elektronske gotovine (Ali je odgovor prisoten in ali je pravilno sestavljen?):

Ne. (14.1)

Da. (14.2)

(13.1)(14.1) Zavrnitev plačila.

(14.2) Potrditev plačila in sprememba (povečanje) stanja e-računa (eR) na napravi.

(15) Pošiljanje potrditve ali zavrnitve plačila.

MA (T – K) (pošiljanje potrditve ali zavrnitve plačila)

(16) Uporabnika združita napravi in se dotakneta zaslona.

MA (K) (zaključek postopka za plačilo)

(17) Sprejem potrditve ali zavrnitve plačila (Ali je bilo plačilo potrjeno?):

Ne. (18.1)

Da. (18.2)

(18.1) Obvestilo uporabniku (zavrnitev). Zaključek postopka za plačilo.

(18.2) Obvestilo uporabniku (potrditev). Sprememba (zmanjšanje) stanja e-računa (eR) na napravi.

Polog elektronske gotovine se izvede z mobilno aplikacijo (MA), katera se preko brezžičnega omrežja poveže s strežniško aplikacijo (SA).

MA (preverjanje vzpostavljene internetne povezave, pošiljanje zahteve za polog in vseh enot elektronske gotovine)

(3.3) Testiranje internetne povezave (Ali ima naprava vzpostavljeno povezavo s internetom?):

Ne. (4.1)

Da: (4.2)

(4.1) Obvestilo uporabniku (internetna povezava ni vzpostavljena).

(5) Pošiljanje zahteve za polog (request: deposit) in vseh enot prejete elektronske gotovine.

SA (testiranje prejete elektronske gotovine, sprememba stanja na transakcijskem računu (tR) in pošiljanje potrditve o pologu)

(6) Testiranje veljavnosti vseh enot prejete elektronske gotovine (Ali je digitalni podpis veljaven?):

Ne. (7.1)

Da. (7.2)

(7.1) Enota elektronske gotovine ni veljavna.

(7.2) Testiranje dvojne porabe za vsako enoto elektronske gotovine (Ali je bila enota elektronske gotovine že porabljena?):

Da. (8.1)

Ne. (8.2)

(8.1) Razkritje identitete uporabnika, ki je izvedel dvojno porabo z enoto elektronske gotovine.

(8.2) Dodajanje enote elektronske gotovine na seznam uspešno položenih.

(9) Sprememba (povečanje) stanja na transakcijskem računu (tR).

(10) Pošiljanje seznama uspešno položenih enot, kot potrditve o pologu.

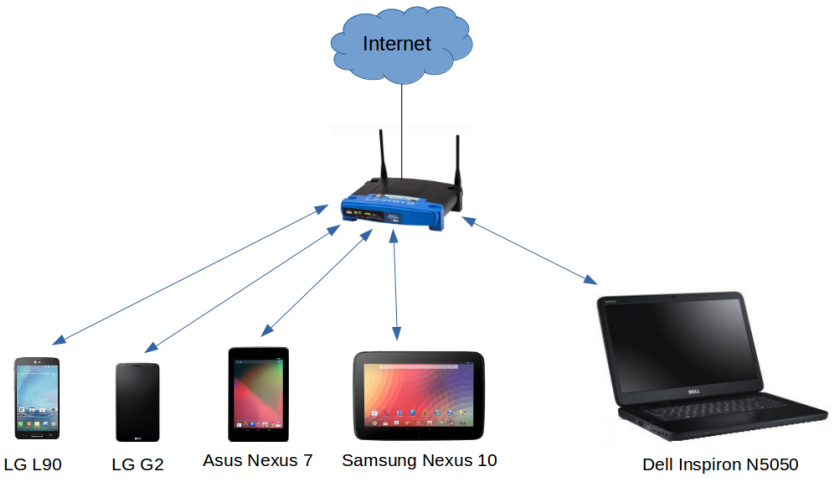
MA (sprememba stanja e-računa (eR) in zaključek postopka za polog)

(11) Sprememba (zmanjšanje) stanja e-računa (eR) na napravi glede na prejeti seznam uspešno položene elektronske gotovine.

(12) Obvestilo uporabniku (postopek za polog zaključen)

5.2 Sistem za brezkontaktno plačevanje

Slika 5.1 prikazuje strojno opremo sistema za testiranje in evaluacijo. Vključuje testni strežnik z nameščeno strežniško aplikacijo in več odjemalcev, mobilnih naprav z nameščeno aplikacijo *NFCePayment*.



Slika 5.1: Strojna oprema testnega sistema za brezkontaktno plačevanje.

Testni strežnik

Strežniška aplikacija, ki je predstavljala bančni sistem, je bila nameščena na prenosnem računalniku tipa Dell Inspiron N5050 z naslednjimi specifikacijami:

- procesor Intel Core i3 (4 x 2.40 GHz),
- pomnilnik 4 GB RAM (DDR2),
- trdi disk SATA 500 GB in
- žična ter brezžična mrežna kartica.

Na njem je bil nameščen operacijski sistem Ubuntu 14.04 LTS (prosto dostopna distribucija Linuxa) in orodje NetBeans IDE 8.0.2 z Javo SDK 7.

Mobilne naprave

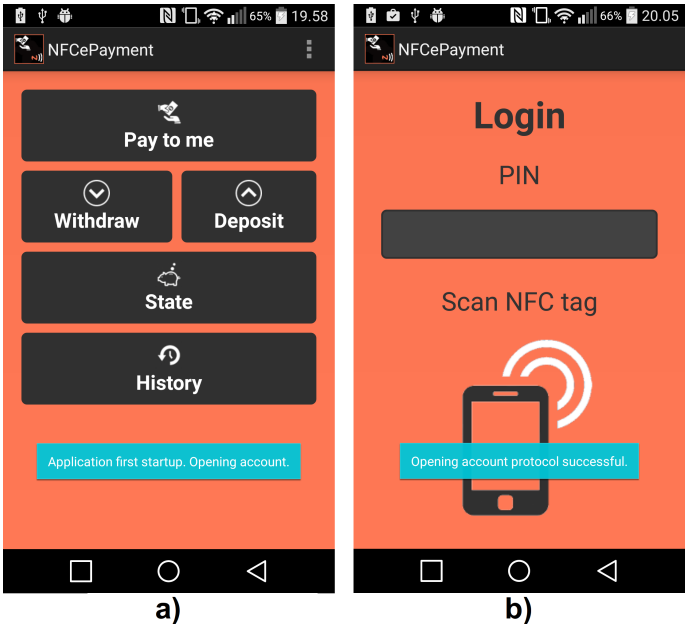
Testiranje mobilne aplikacije *NFCePayment* in celotnega sistema za brezkontaktno plačevanje, je bilo izvedeno z uporabo dveh mobilnih telefonov proizvajalca LG (tip G2 in L90) in enega proizvajalca Samsung (tip Galaxy S5 Mini). Poleg mobilnih telefonov sta bila uporabljena tudi tablična računalnika Asus Google Nexus 7 in Samsung Google Nexus 10. Strojne in programske specifikacije uporabljenih naprav so prikazane v tabeli 5.1.

Tabela 5.1: Specifikacije mobilnih naprav.

naprava	procesor(MHz)	RAM(GB)	NFC	Android
LG G2	Quad c. 2260	2	Da	Lollipop 5.0.2
LG L90	Quad c. 1200	1	Da	Lollipop 5.0.2
Samsung G. S5 M.	Quad c. 1400	1.5	Da	Ice C. S. 4.0.1
Asus Nexus 7	Quad c. 1500	2	Da	Lollipop 5.1.1
Samsung Nexus 10	Dual c. 1700	2	Da	KitKat 4.4.2

5.3 Vzpostavitev sistema

Vzpostavitev testnega sistema je zajemala inicializacijo strežniške aplikacije in postopek za ustvarjanje e-računa (eR) ter namestitev mobilne aplikacije. Najprej smo na testnem strežniku zagnali strežniško aplikacijo in izvedli njeno inicializacijo. Ta je zajemala ustvarjanje para ključev RSA, potrebnega za uspešno delovanje sistema. Aplikacijo *NFCePayment* smo na mobilne naprave namestili iz prenosnega računalnika s pomočjo orodja Android Studio preko povezave USB. Po končani namestitvi se aplikacija samodejno zažene in izvede se preverjanje prisotnosti tehnologije NFC. Če ima mobilna naprava vgrajen modul NFC in vzpostavljeno internetno povezavo, se začne postopek za ustvarjanje e-računa (eR), kar je razvidno iz prikazanega obvestila na sliki 5.2 (a). Med njegovo izvedbo so vse funkcionalnosti aplikacije onemogočene, dokler se postopek uspešno ne zaključi. Po končanem postopku se odpre stran za prijavo in prikaže se obvestilo o uspešno zaključenem postopku (slika 5.2 (b)).



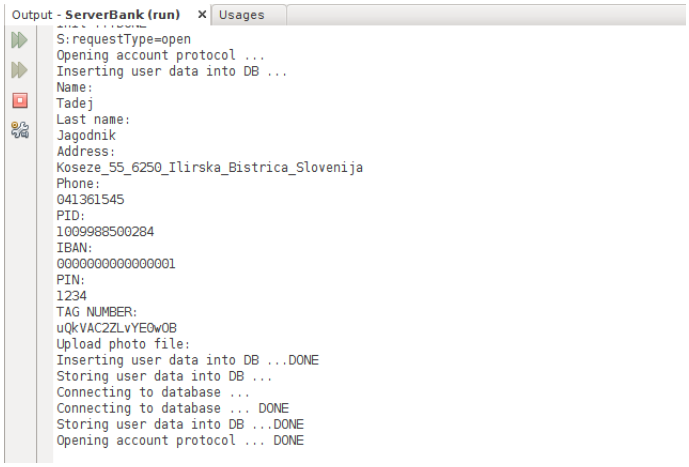
Slika 5.2: Izvedba postopka ustvarjanja e-računa (eR) v mobilni aplikaciji.

V strežniški aplikaciji smo med postopkom za ustvarjanje e-računa (eR) vne-

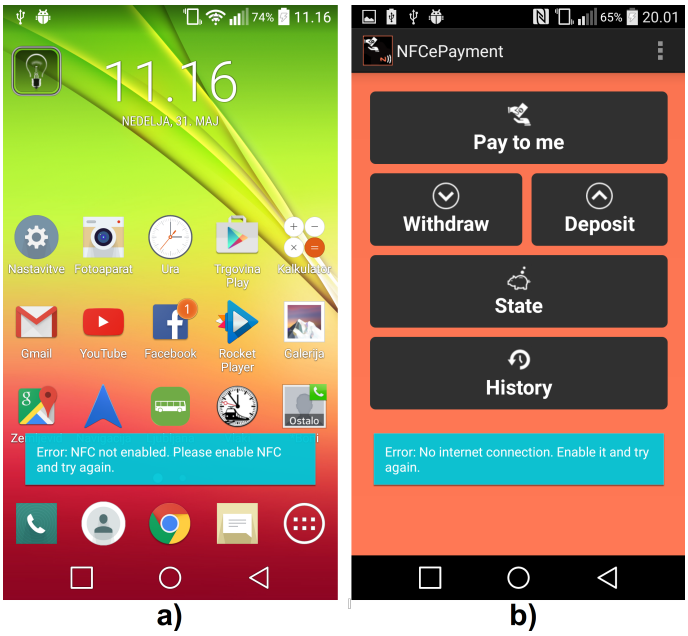
sli podatke o uporabniku, ki so se nato shranili v podatkovno bazo. Uspešen potek in zaključek postopka je prikazan na sliki 5.3.
Med postopkom lahko pride do različnih napak, ki povzročijo njegovo prekinitev. Če naprava nima vgrajenega modula NFC, izvedba postopka in nadaljnja uporaba aplikacije nista mogoči (slika 5.4 (a)). Pred začetkom postopka mora imeti naprava vzpostavljeno internetno povezavo, sicer se uporabniku prikaže ustrezno obvestilo in postopek je potrebno ponoviti (slika 5.4 (b)). Preostale napake, ki se nanašajo na komunikacijo in podatke med postopkom za ustvarjanje e-računa (eR) so:

- napaka pri vzpostavitvi povezave s strežnikom,
- napaka pri pošiljanju podatkov o uporabniku,
- napaka pri shranjevanju podatkov o uporabniku,
- napaka pri pošiljanju javnega ključa bančnega sistema in
- napaka pri shranjevanju javnega ključa.

Za vsako izmed zgornjih napak se prikaže ustrezno obvestilo in postopek ustvarjanja računa (eR) je potrebno ponoviti.



Slika 5.3: Izvedba postopka ustvarjanja e-računa (eR) v strežniški aplikaciji.



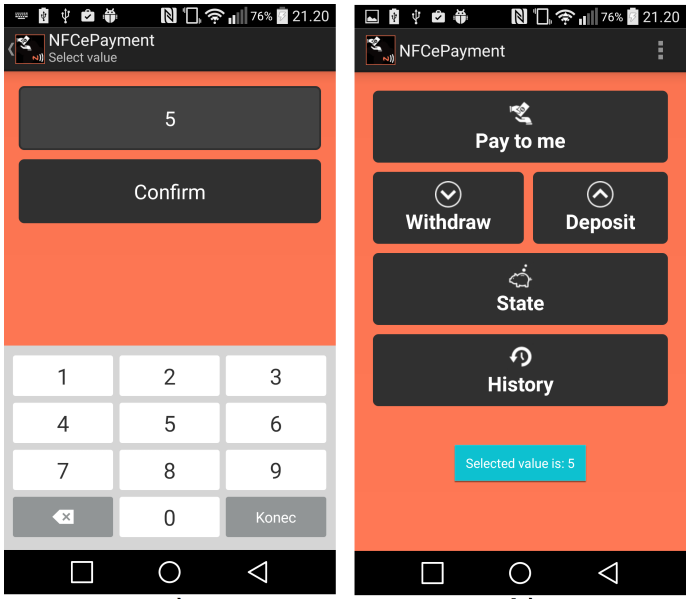
Slika 5.4: Primer obvestila o napaki med ustvarjanje e-računa (eR).

5.4 Uporaba elektronske gotovine

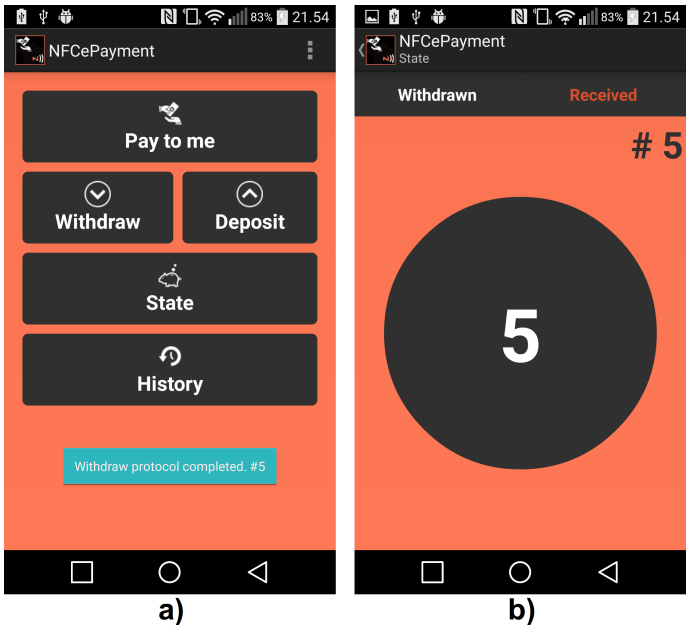
5.4.1 Dvig

Postopek za dvig elektronske gotovine pričnemo s klikom na gumb Dvig (Withdraw), nakar sledi preverjanje ali ima naprava vzpostavljeno internetno povezavo, potrebno za uspešno izvedbo postopka. Če je preverjanje uspešno, se odpre stran za vnos zneska (slika 5.5 (a)). V vnosno polje smo s pomočjo številske tipkovnice vnesli vrednost pet in kliknili na gumb Potrdi (Confirm). S tem se prične postopek za dvig petih enot elektronske gotovine med mobilno in strežniško aplikacijo, kar je razvidno iz obvestila prikazanega na sliki 5.5 (b). Po uspešno zaključenem postopku se prikaže obvestilo na sliki 5.6 (a), ki vključuje tudi trenutno stanje dvignjene elektronske gotovine na e-računu (eR). O uspešnosti dviga se lahko dodatno prepričamo tudi s preverjanjem strani, ki prikazuje stanje dvignjene elektronske gotovine na e-računu (eR) in je prikazana na sliki 5.6 (b).

Uspešen potek postopka za dvig ene enote elektronske gotovine v strežniški

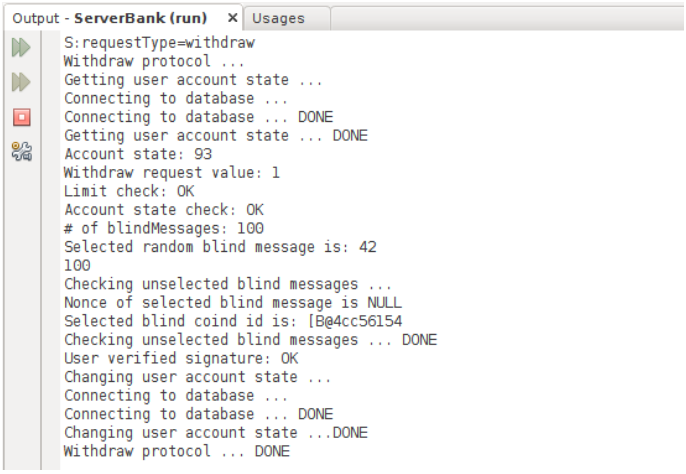


Slika 5.5: Izvedba dviga elektronske gotovine v mobilni aplikaciji (1).



Slika 5.6: Izvedba dviga elektronske gotovine v mobilni aplikaciji (2).

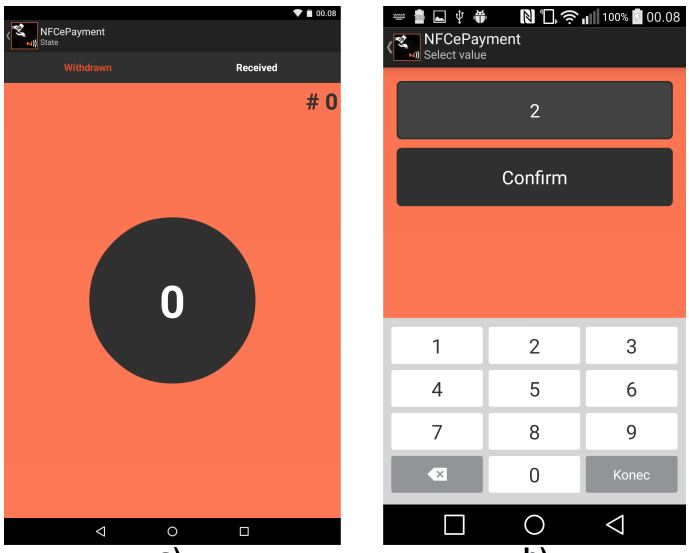
aplikaciji je prikazan na sliki 5.7.



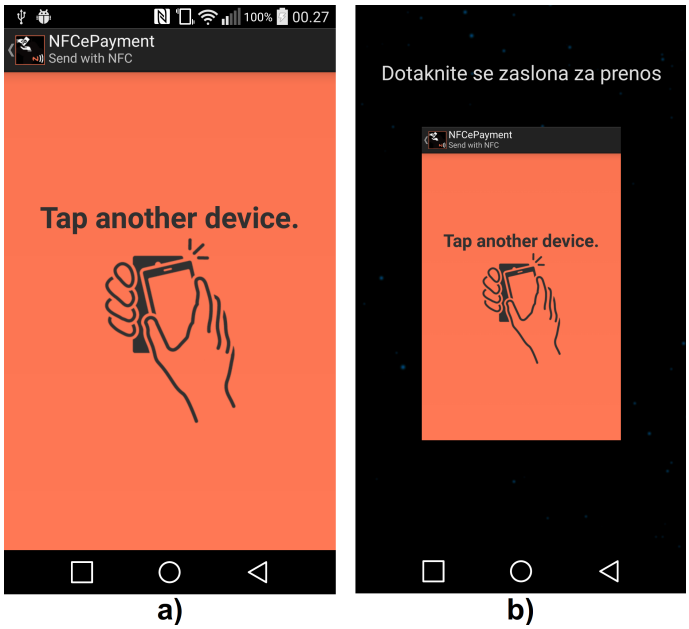
Slika 5.7: Izvedba dviga elektronske gotovine v strežniški aplikaciji.

5.4.2 Plačilo

Za izvedbo brezkontaktnega plačila smo uporabili mobilni napravi plačnika in prejemnika plačila. Izvede se v treh korakih in zajema prav toliko prenosov podatkov s pomočjo tehnologije NFC. Pred začetkom plačila smo na napravi prejemnika plačila preverili stran, ki prikazuje stanje prejete elektronske gotovine (vrednost 0) (slika 5.8 (a)). Postopek za plačilo začnemo tako, da na napravi prejemnika plačila kliknemo na gumb Plačaj mi (Pay to me), nakar se izvede preverjanje ali je modul NFC vključen in pripravljen na uporabo. Če je preverjanje uspešno, se odpre stran za vnos zneska (slika 5.8 (b)). V vnosno polje smo s pomočjo številske tipkovnice vnesli vrednost 2 in kliknili na gumb Potrdi (Confirm) ter tako ustvarili zahtevek za plačilo.
Pošiljanje z uporabo tehnologije NFC na mobilnih napravah Android, poteka s pomočjo funkcije Android Beam tako, da napravi združimo. Navodilo, ki prejemniku plačila pove, da je za prenos potrebno združiti napravi, je prikazano na sliki 5.9 (a). Ko sta napravi združeni, se je potrebno za začetek in potrditev prenosa dotakniti zaslona obeh naprav hkrati (slika 5.9 (b)).

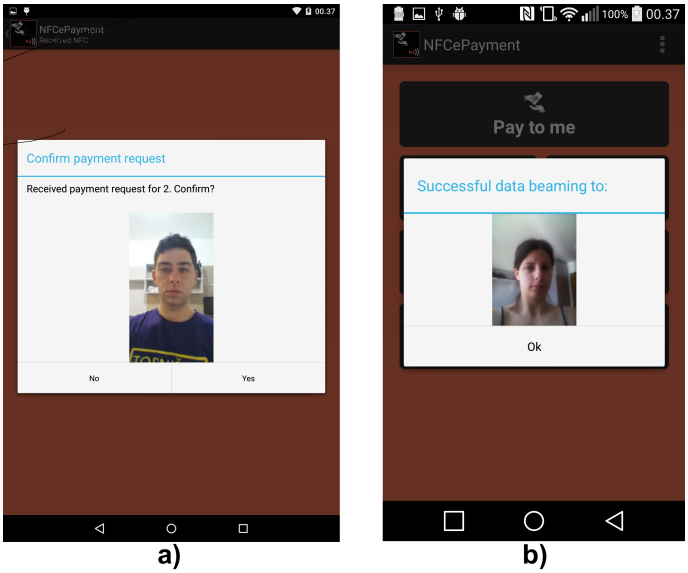


Slika 5.8: Izvedba brezkontaktnega plačila v mobilni aplikaciji (1): prejemnik plačila.



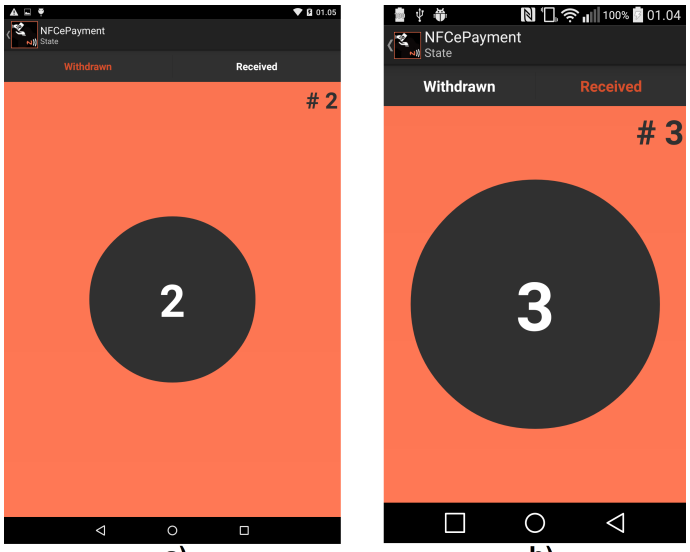
Slika 5.9: Izvedba brezkontaktnega plačila v mobilni aplikaciji (2): prejemnik plačila.

Po končanem pošiljanju zahtevka, se nam na napravi prejemnika plačila prikaže obvestilo v obliki pogovornega okna o uspešnem prenosu, ki vsebuje tudi fotografijo plačnika (slika 5.10 (b)). Pogovorno okno smo zaprli s klikom na gumb V redu (Ok). Na napravi plačnika se odpre pogovorno okno, ki vsebuje znesek zahtevanega plačila, fotografijo pošiljatelja (prejemnika plačila) in možnost potrditve ali zavrnitve plačila (slika 5.10 (a)).



Slika 5.10: Izvedba brezkontaktnega plačila v mobilni aplikaciji (3): a) prejemnik plačila, b) plačnik.

Plačilo smo potrdili s klikom na gumb Da (Yes). Nato smo za prenos podatkov o plačilu ponovno združili napravi in se dotaknili njunih zaslonov na enak način kot prej.
Po prejetem plačilu se na napravi prejemnika plačila izvede preverjanje veljavnosti prejete elektronske gotovine. Nato sledi še zadnji prenos NFC, ki zajema potrditev ali zavrnitev plačila s strani prejemnika plačila.
Po končanem plačilu smo preverili stanje prejete elektronske gotovine na napravi prejemnika plačila (slika 5.11 (a)) in stanje dvignjene elektronske gotovine, ki je bila zmanjšana za znesek plačila na strani plačnika (slika 5.11 (b)).

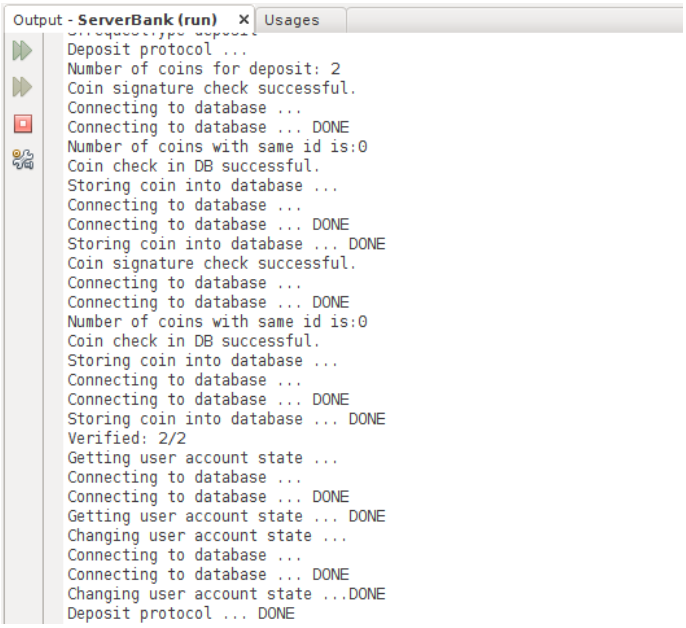


Slika 5.11: Izvedba brezkontaktnega plačila v mobilni aplikaciji (4): a) prejemnik plačila, b) plačnik.

5.4.3 Polog

Po opravljenem brezkontaktnem plačilu imamo na e-računu (eR) prejemnika plačila shranjeni dve enoti elektronske gotovine, kar je razvidno iz slike 5.11 (a). Če jo želimo unovčiti je potrebno izvesti postopek za polog, ki ga začnemo tako, da kliknemo na gumb Položi (Desposit), nato sledi preverjanje ali ima naprava vzpostavljeno internetno povezavo. Če je uspešno, se začne postopek za prenos podatkov, kar je razvidno iz obvestila (Deposit prtocol started). Mobilna aplikacija strežniški posreduje vso prejeto elektronsko gotovino iz e-računa (eR). Po zaključenem postopku se prikaže obvestilo o opravljenem pologu, ki vključuje tudi število posredovanih in uspešno položenih enot elektronske gotovine.

Na strežniški aplikaciji se med postopkom za polog izvede preverjanje elektronske gotovine, posredovane s strani mobilne aplikacije. Če je preverjanje uspešno, se v podatkovno bazo doda položeno elektronsko gotovino in spremeni uporabnikovo stanje transakcijskega računa (tR). Uspešen potek in zaključek postopka v strežniški aplikaciji je prikazan na sliki 5.12.



Slika 5.12: Izvedba postopka za polog v strežniški aplikaciji.

5.5 Napake

Zaradi uporabe različnih tehnologij in podatkov, ki se uporabljajo, se lahko pojavijo določene napake v kateremkoli od opisanih postopkov. V tem primeru se postopek prekine in pojavijo se ustrezni opisi napak.

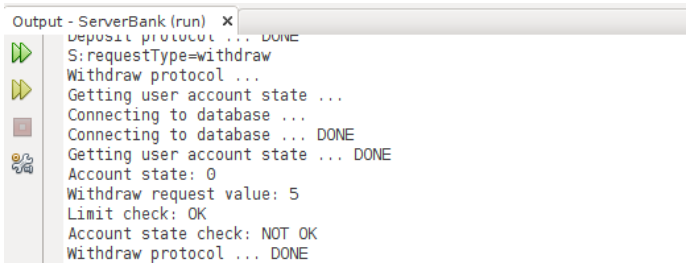
5.5.1 Dvig in polog

Pred začetkom postopkov za dvig in polog mora imeti naprava vzpostavljeno internetno povezavo, sicer se prikaže obvestilo o napaki (slika 5.4 (b)). Prav tako lahko pride do napake pri vzpostavitvi povezave s strežnikom.
Napake, ki se lahko zgodijo med postopkom za dvig in se nanašajo na komunikacijo in podatke so:

- napaka pri vnosu zneska manjšega ali enakega nič,
- napaka zaradi prevelikega zneska dviga,
- napaka zaradi nezadostnega stanja na transakcijskem računu (tR),

- napaka pri pridobivanju shranjenega javnega ključa,
- napaka pri pridobivanju shranjenih podatkov o uporabniku,
- napaka pri pošiljanju parametra n,
- napaka pri preverjanju slepih sporočil,
- napaka pri preverjanju digitalnega podpisa in
- napaka pri shranjevanju elektronske gotovine na e-račun (eR).

Postopek neuspešne izvedbe postopka za dvig, zaradi nezadostnega stanja bančnega računa uporabnika v strežniški aplikaciji je prikazan na sliki 5.13.



Slika 5.13: Izvedba neuspešnega postopka za dvig v strežniški aplikaciji.

Napake, ki se lahko zgodijo med pologom in se nanašajo na komunikacijo in podatke so:

- napaka pri poskusu pridobitve prejete elektronske gotovine z e-računa (eR),
- napaka pri pošiljanju elektronske gotovine za polog,
- napaka pri pošiljanju potrditve o položitvi elektronske gotovine in
- napaka, pri odstranjevanju elektronske gotovine iz e-računa (eR).

Do napak lahko pride tudi med preverjanjem položene elektronske gotovine, ki se izvede v strežniški aplikaciji. Za vsako posamezno enoto elektronske

gotovine se preveri njen digitalni podpis in dvojno porabo. V primeru neuspešnega preverjanja se po končanem postopku uporabniku prikaže obvestilo, kjer se število posredovanih in uspešno položenih enot elektronske gotovine ne ujema (Deposit protocol completed. #1/0).

5.5.2 Plačilo

Pred pričetkom prenosa morata obe napravi imeti vključen modul NFC, sicer se prikaže obvestilo o napaki. Nato mora naprava vključiti možnost uporabe modula NFC in postopek se lahko ponovi. Pri prenosu s pomočjo funkcije Android Beam lahko tudi pride do napake in prikaže se obvestilo, ki pove uporabnikoma, da morata znova združiti naprave. V tem primeru je potrebno za ponovitev prenosa napravi ponovno združiti. Napake, ki se nanašajo na izvedbo plačila so:

- napaka pri preverjanju stanja elektronske gotovine plačnika,
- napaka pri pridobivanju dvignjene elektronske gotovine iz e-računa(eR),
- napaka pri odstranjevanju dvignjene elektronske gotovine iz e-računa (eR),
- napaka pri preverjanju prejete elektronske gotovine in
- napaka pri shranjevanju prejete elektronske gotovine na e-račun (eR).

5.6 Dvojna poraba

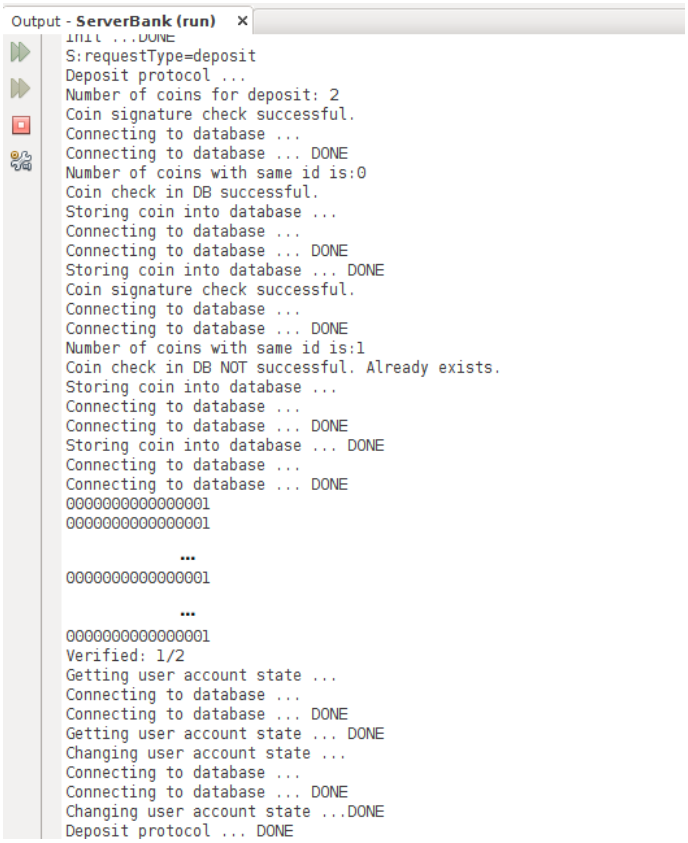
Preverjali smo tri načine dvojne porabe elektronske gotovine, ki jo je mogoče izvesti z mobilno aplikacijo:

- plačnik ne odstrani uporabljene elektronske gotovine na svoji napravi,
- prejšnjemu primeru plačnik spremeni identifikator elektronske gotovine in

- prejemnik plačila dvakrat izvede polog.

Za prvo testiranje dvojne porabe smo najprej na napravi plačnika izvedli postopek za dvig ene enote elektronske gotovine. Pred izvedbo postopka za plačilo smo na isti napravi onemogočili odstranjevanje dvignjene elektronske gotovine iz e-računa (eR). Nato smo dvakrat izvedli postopek za brezkontaktno plačilo ene enote elektronske gotovine. Po opravljenih plačilih je bilo stanje dvignjene elektronske gotovine na e-računu (eR) plačnika še vedno enako ena, na e-računu (eR) prejemnika plačila pa dva. Na koncu smo na mobilni napravi prejemnika plačila izvedli postopek za polog dveh enot elektronske gotovine.
V strežniški aplikaciji se izvede preverjanje veljavnosti in možnosti dvojne porabe, ki se izvede za vsako enoto posebej. Preverjanje prve enote elektronske gotovine je uspešno in strežniška aplikacija jo shrani v podatkovno bazo. Pri preverjanju druge enote, pa bančni sistem zazna dvojno porabo, saj je bila enota z istim identifikatorjem že predhodno položena (slika 5.14). Strežniška aplikacija na podlagi obeh enot elektronske gotovine, ugotovi identiteto storilca dvojne porabe (iban številka 0000000000000001), kar je razvidno iz slike 5.14. Po končanem postopku se na napravi prejemnika plačila prikaže obvestilo o številu uspešno položenih enot elektronske gotovine, kar je razvidno tudi pri pregledu stanja prejete elektronske gotovine, ki je enako ena.
Pri drugem testiranju dvojne porabe smo ponovili zgornji postopek, le pred izvedbo drugega brezkontaktnega plačila smo enoti elektronske gotovine še dodatno spremenili identifikator. Med izvedbo plačila se na napravi prejemnika plačila izvede preverjanje posredovane enote elektronske gotovine, ki vključuje tudi preverjanje veljavnosti elektronske gotovine z javnim ključem RSA bančnega sistema. Ker preverjanje digitalnega podpisa ni bilo uspešno se postopek brezkonatktnega plačila ni zaključil in tudi do izvedbe postopka za polog ni prišlo.
Pri zadnjem testiranju dvojne porabe smo najprej na napravi plačnika izvedli postopek za dvig ene enote elektronske gotovine in nato še postopek za brezkontaktno plačilo. Po opravljenem plačilu je bilo stanje dvignjene

elektronske gotovine na e-računu (eR) plačnika enako nič, na e-računu (eR) prejemnika plačila pa ena. Na napravi prejemnika plačila smo onemogočili odstranjevanje prejete elektronske gotovine in dvakrat izvedli postopek za polog. Pri prvi izvedbi postopka je preverjanje enote elektronske gotovine uspešno, pri drugi pa strežniška aplikacija zazna dvojno porabo. Ker sta izziva in odgovora obeh enot elektronske gotovine enaka, strežniška aplikacija ugotovi identiteto storilca dvoje porabe, prejemnika plačila.



Slika 5.14: Izvedba neuspešnega postopka za polog elektronske gotovine (dvojna poraba) v strežniški aplikaciji.

5.7 Analiza

Analiza sistema za brezkontaktno plačevanje vključuje pregled značilnosti tehnologije NFC, rezultate testiranja, varnost podatkov in njihovo izmenjavo na mobilnih Android napravah ter potencialne možnosti uporabe rešitve.

SWOT (Strenghts, Weaknesses, Opportunities, Threats) je ena izmed najpogosteje uporabljenih strukturiranih oblik vrednotenja projektov, programov, izdelkov ali storitev [12]. V njej je potrebo podrobno analizirati naslednje štiri aspekte:

- prednosti (angl. strenghts),
- slabosti (angl. weaknesses),
- priložnosti (angl. opportunities) in
- nevarnosti (angl. threats).

Prednosti in slabosti se nanašata na notranje dejavnike, kar pomeni, da imamo na njiju neposreden vpliv. Lahko jih urejamo, spreminjamo, preoblikujemo ali ukrepamo na kakšen drugi način. Priložnosti in nevarnosti pa se nanašajo na zunanje dejavnike. Na njih nimamo neposrednega vpliva in se jim lahko zgolj prilagodimo. Prednosti predstavijo dejavnike, ki pozitivno vplivajo na doseganje določenega cilja. Slabosti razkrijejo šibkosti analiziranega, ki jih je še potrebno popraviti, dodelati ali spremeniti. Potrebno je razlikovati različne vrste slabosti in katere imajo večji ali manjši pomen. Na priložnosti nimamo posebnega vpliva, a jih lahko izkoristimo sebi v prid. S tem omogočimo analiziranemu, da doseže določene pozitivne učinke. Nevarnosti so najbolj pereč del analize, saj gre za dejavnike, ki imajo negativen učinek in na njih ne moremo vplivati. Bistvenega pomena je, da skušamo predvideti negativne dejavnike iz okolja ter naredimo načrt, kako bomo v tem primeru ravnali.

Za tehnologijo NFC je analiza SWOT že izdelana in jo je mogoče najti v različnih virih. V tabeli 5.2 smo povzeli tiste značilnosti, ki so pomembne za delovanje našega sistema:

Tabela 5.2: Tabela SWOT za tehnologijo NFC.

Prednosti	**Slabosti**
- Delovanje na mobilnih napravah - Velik delež mobilnih naprav NFC - Uporaba mobilnih naprav vedno in povsod (vseprisotnost) - Veliko število aplikacij, ki uporabljajo NFC - Priročnost in enostavnost uporabe NFC - Izvedba komunikacije NFC na interaktiven način - Hitrost izvedbe prenosa podatkov - Uporaba obstoječe infrastrukture	- Dražje naprave NFC - Zmanjševanje življenjske dobe baterije - Vprašanje na področju varnosti in zasebnosti - Varnost operacijskega sistema pred zlonamerno programsko opremo - Omejena uporaba
Priložnosti	**Nevarnosti**
- Nova in inovativna tehnologija - Možnosti razvoja novih rešitev - Možnosti odprtja novih trgov - Praktična uporaba - Razvoj varnostnih mehanizmov	- Pomanjkanje zakonskih predpisov - Konkurenčnost ostalih tehnologij

Prednosti

Mobilne naprave so na voljo vse večjemu številu ljudi, ki jih nosijo vedno s seboj in jih povsod uporabljajo. Poleg tega se povečuje delež mobilnih naprav, ki imajo poleg ostalih funkcionalnosti vključeno tudi tehnologijo NFC. Na področju tehnologije NFC obstaja veliko število standardiziranih rešitev, ki so kompatibilne z mobilnimi napravami, kar olajša njeno uporabo. Slednje se odraža v velikem številu aplikacij, ki uporabljajo tehnologijo NFC. Ker je ta brezžična je primerna za izmenjavo podatkov, njena izvedba na kratki razdalji daje uporabniku večji občutek varnosti. Uporaba tehnologije NFC na mobilnih napravah je za uporabnika priročna in enostavna. Izvedba

komunikacije NFC je interaktivna, saj je potrebno za njeno izvedbo združiti napravi NFC.

Slabosti

Slabost mobilnih naprav z vključeno tehnologijo NFC je njihova cena, saj so te v določeni meri dražje od preostalih. Zato se ljudje odločajo za nakup cenovno ugodnejših naprav. Čeprav so mobilne naprave vedno bolj zmogljivejše, uporaba tehnologije NFC zmanjšuje življenjsko dobo baterije naprave. Iz področju varnosti in zasebnosti obstaja še veliko odprtih vprašanj, ki se nanašajo na komunikacijo NFC in podatke, ki se prenašajo. Pomembna je tudi varnost operacijskega sistema, ki tehnologijo NFC uporablja, predvsem pred zlonamerno programsko operemo. Slabost predstavlja tudi njena uporaba na mobilnih napravah, kjer niso podprte vse funkcionalnosti tehnologije, zato je njena uporaba omejena.

Priložnosti

Vključenost tehnologije NFC v mobilne naprave prinaša možnosti za razvoj novih inovativnih rešitev. Odpira se veliko novih trgov, predvsem na področju mobilnih aplikacij za plačevanje. Obstoječa infrastruktura in veliko število mobilnih naprav z vključeno tehnologijo NFC omogočata izvedbo in testiranje rešitve v praksi. Vse pogostejša uporaba tehnologije NFC odpira veliko možnosti za razvoj novih in inovativnih varnostnih mehanizmov. Uporaba tehnologije NFC v obstoječih rešitvah in sistemih izboljša vključenost uporabnika v samih sistemih. Prav tako se poveča interaktivnost med uporabnikom in sistemom.

Nevarnosti

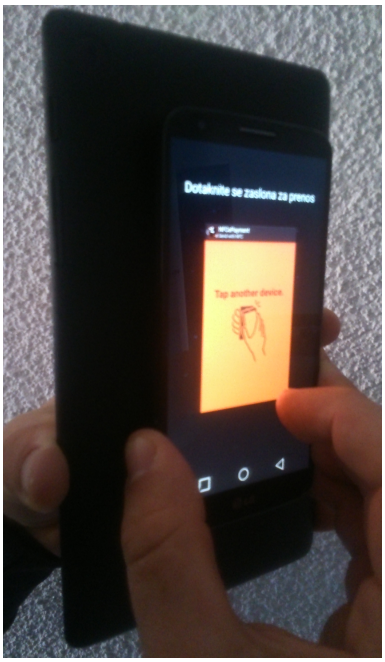
Nevarnost predstavlja predvsem pomanjkanje zakonskih predpisov in pravil na področju tehnologije NFC in njene uporabe na mobilnih napravah. Poleg tega se tehnologija srečuje s konkurenco ostalih komunikacijskih tehnologij, ki so že uveljavljene.

Uporaba aplikacije *NFCePayment*, ki omogoča izvedbo brezkontaktnega plačila, ima enostaven uporabniški vmesnik, ki omogoča hitro izbiro aktivnosti in navigacijo med njimi. Uporabniški vmesnik je za vse mobilne naprave zasnovan na enak način, vendar je do določene mere prilagojen na velikosti njenega zaslona. Uporabniška izkušnja je veliko boljša na napravah z večjimi zasloni (tablici), saj je uporaba aplikacije hitrejša in predvsem učinkovitejša. Večja velikost gumbov in vnosnih polj uporabniku omogoča hitrejši interakcijo z mobilno napravo in aplikacijo. Prav tako so na napravah z večjimi zasloni prikazana večja obvestila in s tem bolj pregledna.

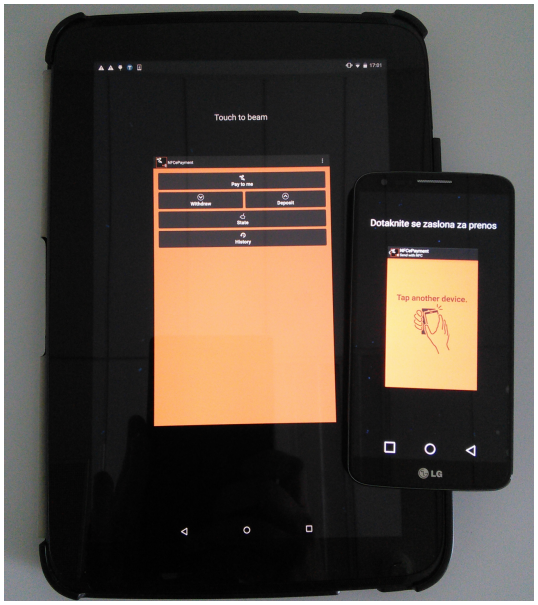
Velikost mobilne naprave vpliva tudi na njeno prenašanje. Naprave z večjim zaslonom so težje prenosljive kot tiste z manjšim, zato so bolj primerne za prodajalce (prejemnike plačila). Ti jih imajo lahko postavljene na prodajnih pultih, kjer kupec (plačnik) izvede plačilo. Medtem, ko so manjše naprave bolj primerne za kupce, saj jih lahko enostavno prenašajo in jim predstavljajo nekakšno virtualno denarnico.

Uporabniška izkušnja med izvedbo brezkontaktnega plačila s pomočjo tehnologije NFC, se prav tako razlikuje glede na tip uporabljenih mobilnih naprav. Odvisna je od fizične lokacije modula NFC na napravi, same velikosti naprave in različice operacijskega sistema Android.
Mobilne naprave, kot so telefoni in tablice z manjšimi zasloni (npr. Asus Nexus 7) imajo modul NFC vgrajen na zadnji strani. Pri uporabi tovrstnih naprav ju je za izvedbo prenosa NFC potrebno združiti tako, da se njuni zadnji strani dotikata (slika 5.15). Mobilne naprave, predvsem tablice z večjimi zasloni (npr. Samsung Nexus 10) pa imajo modul NFC vgrajen na sprednji stran. Prenos NFC se izvede tako, da drugo napravo položimo na sprednjo stran tablice (slika 5.16). Slednje močno olajša izvedbo brezkotaktnega plačila in izboljša nejgovo uporabniško izkušnjo. Predvsem se izognemo nerodnemu združevanju naprav z modulom NFC vgrajenim na zadnji strani.

Tablica je postavljena na prodajnem pultu, kjer kupec izvede plačilo tako, da na njo položi svojo mobilno napravo NFC.
Uporabniška izkušnja v primeru izvedbe brezkontaktnega plačila med dvema



Slika 5.15: Prenos NFC med tablico z vgrajenim modulom NFC na zadnji strani in telefonom.



Slika 5.16: Prenos NFC med tablico z vgrajenim modulom NFC na sprednji strani in telefonom.

naparavam z vgrajenim modulom NFC je odvisna tudi od njene velikosti, saj je prijem večjih naprav (npr. Asus Nexus 7) med združevanjem otežen. Uporaba funkcije Android Beam ima dodan del uporabniškega vmesnika, kjer se mora uporabnik dotakniti svojega zaslona naprave za izvedbo prenosa. Slednje poslabša uporabniško izkušnjo brezkontaktnega plačila, saj se morata uporabnika dotakniti svojih zaslonov skoraj istočasno, sicer je lahko prenos neuspešen in ga je potrebno ponoviti. V novejših različicah operacijskega sistema Android (od Android 5.0. naprej) je ta del odstranjen in uporabnika morata za dokončanje prenosa le združiti naprave, kar odstrani odvečni korak dotikanja zaslonov in pospeši izvedbo plačila. Uporabniška izkušnja se sicer izboljša, vendar je njena implementacija za enkrat omejena, kar prinaša dodatne težave, ki vplivajo na delovanje aplikacije.

Ker je izvedba brezkontaktnega plačila izvedena s pomočjo treh prenosov NFC, je čas njegove izvedbe odvisen od uspešnosti prenosov. V primeru uspešnih prenosov je potek plačila relativno hiter, v primeru napak med prenosom pa se njegov čas močno poveča.

Prijava v aplikacijo je izvedena s pomočjo kode PIN in značke NFC. Uporabnik se lahko prijavi v aplikacijo le ob poznavanju kode PIN in posedovanju značke NFC, sicer prijava ni mogoča. Uporabniški vmesnik prijave je zasnovan tako, da uporabnik vnese kodo PIN in nato približa svojo značko modulu NFC. Po uspešnem branju vsebine iz značke se izvede preverjanje uspešnosti prijave, kar nadomesti uporabo gumba za potrditev. Ker uporabniku ni potrebno poleg vpisa kode PIN in branja značke NFC storiti ničesar drugega prijava poteka hitro.

Poglavje 6

Sklepne ugotovitve

V magistrskem delu je bil razvit sistem, ki omogoča izvedbo brezkontaktnega plačevanja. Namenjen je demonstraciji uporabe elektronske gotovine, ki je shranjena na mobilnih napravah NFC. Izvaja se z mobilno aplikacijo *NFCePayment* v operacijskem sistemu Android. Implementirana je tudi strežniška aplikacija, ki je namenjena testiranju in evalvaciji sistema. Sistem omogoča anonimno izmenjavo elektronske gotovine v nepovezanem načinu s pomočjo komunikacije P2P v tehnologiji NFC, ki močno spominja na obstoječi gotovinski način. V implementaciji so vključeni trije udeleženci (banka, plačnik in prejemnik plačila). Definirani so postopki za ustvarjanje e-računa (eR), dvig, brezkonatktno plačilo in polog. Arhitektura sistema je razdeljena na strežniško aplikacijo, ki predstavlja bančni sistem in mobilno aplikacijo. Dvofaktorska identifikacija je v mobilni aplikaciji izvedena z uporabo kode PIN in kartice NFC. Dvojna poraba in veljavnost elektronske gotovine pa se preverjata pri pologu v strežniški aplikaciji.

Število pametnih mobilnih naprav, ki imajo poleg drugih funkcionalnosti vključeno tudi tehnologijo NFC, se iz dneva v dan veča. Tako postajajo vse bolj uporabne in priročne, kar se odraža v velikem številu različnih aplikacij. Mobilne naprave NFC bi lahko nadomestile uporabo obstoječih oblik plačevanja. S tem bi se izognili potrebi po prenašanju gotovine ali plačilnih kartic, saj bi mobilno napravo lahko uporabljali kot virtualno denarnico,

zmanjšali pa bi se tudi stroški plačevanja. Predvsem bi sistem za brezkontaktno plačevanje lahko nadomestil izmenjavo manjših zneskov gotovine, saj omogoča izmenjavo anonimne elektronske gotovine med dvema osebama.

Sistem za brezkontaktno plačevanje bi bil primeren za izvedbo plačil z manjšimi zneski. Ker se plačilo izvede v nepovezanem načinu, je sistem primeren za plačevanje na mestih, kjer ni na voljo mobilnega omrežja ali ni možno vzpostaviti povezave z internetom preko brezžičnega omrežja (podzemne železnice itd.). Ker uporaba elektronske gotovine zahteva izvedbo dviga in pologa v povezavi s strežniško aplikacijo, bi uporabnik lahko slednja izvedel pred oziroma po opravljenem plačilu.

V sistemu lahko z elektronsko gotovino plačilo izvede le oseba, ki je izvedla dvig. Da bi bil plačilni sistem čimbolj podoben gotovinskemu in za izboljšanje njegove učinkovitosti, je smiselna uvedba prenosljivosti elektronske gotovine. V tem primeru bi lahko prejemnik plačila prejeto elektronsko gotovino uporabil naprej kot plačnik, brez predhodne povezave z banko in izvedbe postopka za polog ter dvig. Prav tako bi bila smiselna uporaba enot elektronske gotovine različnih vrednosti, kot pri gotovini. Sistem trenutno uporablja le enote elektronske gotovine v vrednosti ena in je zato precej kompleksen in dolgotrajen.
Omejitev sistema predstavlja tudi obsežen postopek za dvig ene enote elektronske gotovine. Ta je odvisen od vrednosti parametra n. Z zmanjšanjem njegove vrednosti se izboljša učinkovitost, vendar se zmanjšuje varnost sistema, saj se poveča verjetnost goljufanja med izvedbo postopka.
Omenjeni nadgradnji bi lahko izvedli z implementacijo kompleksnejše sheme za plačevanje z izboljšanim postopkom za dvig elektronske gotovine in s podporo tako prenosljivosti kot deljivosti elektronske gotovine.
Pri uporabi različnih vrednosti elektronskih enot, bi bančni sistem lahko uporabljal več različnih parov ključev RSA, za vsako vrednost svojega. S tem bi se lahko izognil možnosti goljufanja med dvigom. Tako bi bila dejanska vrednost elektronske enote odvisna od tega, s katerim ključem jo je banka digitalno podpisala.

Brezkontaktno plačilo se izvede s pomočjo treh prenosov NFC, kar precej podaljša njegovo izvedbo. Razlog je v omejitvah, ki jih prinaša implementacija funkcije Android Beam. Ta ne omogoča izvedbe klasične komunikacije P2P, kjer si napravi na podlagi vzpostavljene komunikacije zaporedno izmenjujeta sporočila, ampak omogoča le pošiljanje enega sporočila NDEF. V operacijskem sistemu Android trenutno ni podprte rešitve, ki bi omogočala izvedbo komunikacije P2P med dvema mobilnima napravama tako, da bi se lahko plačilo izvedlo s pomočjo enega združevanja naprave. Izvedba tovrstne komunikacije je sicer možna na nestandarden način, s spremembo načina delovanja emulacije kartice in bralnega/pisalnega načina [32]. V tem primeru se najprej ena mobilna naprava obnaša kot čitalec NFC, druga pa kot značka NFC. Nato se njuni vlogi lahko obrneta, kar omogoča izvedbo dvosmerne P2P komunikacije.

Predlagan sistem brezkontaktnega plačevanja s tehnologijo NFC v nepovezanem načinu predstavlja enega od konceptov, kjer bi gotovino prenesli na pametne mobilne naprave. Prikazane so bile nekatere prednosti in pomanjkljivosti, ki pa ne prestavljajo ovire za nadaljnje možnosti razvoja podobnih rešitev.

Slike

Literatura

[1] N. Asokan et al., "The state of the art in electronic payment systems", Computer 30.9, str.: 28–35, 1997.

[2] R. Canetti, H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels.", Advances in Cryptology, EUROCRYPT 2001, 2001.

[3] D. Chaum, "Achieving Electronic Privacy", Scientific American, str.: 96–101, 1992.

[4] D. Chaum, "Blind Signatures for Untreaceable Payments", Advances in Cryptology: Proc. CRYPTO 82, New York: Plenum, str.: 199–203, 1983.

[5] D. Chaum, A. Fiat, M. Naor, "Untreaceble Electronic Cash", Advances in in Cryptology: CRYPTO 88, Springer-Verlag, str.: 319–327, 1988.

[6] V. Coscun, K. Ok, B. Ozdenizci, "Near Field Communication From Theory to Practice", Chichester: Wiley, 2012.

[7] V. Coscun, K. Ok, B. Ozdenizci, "Professional NFC Application Development for Android", Chichester: Wiley, 2013.

[8] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication", Chichester: Wiley, 2010.

[9] A. Jules, "RFID Security and Privacy", IEEE Journal on Selected Areas in Communications, st. 24, zv. 2, str.: 381–394, Bradford, USA, 2006.

[10] L. Law, S. Sabett, J. Solinas, "How to make a mint: the cryptography of anonymous electronic cash", Am. UL Rev., 46: 1131, 1996.

[11] L. Mainetti, L. Patrono, R. Vergallo, "IDA-Pay: An innovative micro-payment system based on NFC technology for Android mobile devices", SoftCOM, Split, 2012.

[12] F. Mehmooda, M. Hassannezhada, T. Abbasb, "Analytical investigation of mobile NFC adaption with SWOT-AHP", The 7th International Conference Interdisciplinarity in Engineering (INTER-ENG 2013), str.:535–541, 2013.

[13] O. Ohta, K. Ohta, "Universal Electronic Cash.", Advances in Cryptology, Proceedings Springer–Verlag, str.: 324–337, 1992.

[14] D. O'Mahony, M. Peirece, H. Tewari, "Electronic Payment Systems for E-Commerce", 2nd ed., Artech House, 2001.

[15] B. Schneier, "Applied cryptography: protocols, algorithms, and source code in C.", John Wiley & Sons, 2nd edition, 1996.

[16] U. Trottmann, "NFC-Possibilities and Risks", Network 35, 2013.

[17] T. Dierks, E. Rescorla, "RFC5246: The Transport Layer Security (TLS) Protocol.", Technical report, 2008.

[18] ECMA International, "Standard ECMA-340: Near Field Communication Interface and Protocol (NFCIP-1)", 2013.

[19] ECMA International, "Standard ECMA-352: Near Field Communication Interface and Protocol (NFCIP-2)", 2013.

[20] ECMA International, "Standard ECMA 373: Near Field Communication Wired Interface (NFC-WI)", 2006.

[21] ETSI, "ETSI TS 102 190: Near Field Communication (NFC) IP-1, Interface and Protocol (NFCIP-1), Tehnical Specification" 2003.

[22] ETSI, "ETSI TS 102 312: Near Field Communication Interface and Protocol 2 (NFCIP-2). Tehnical Specification", 2004.

[23] ETSI TS, "ETSI TS 102 613, Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics", Tehnical Specification, 2008.

[24] Google Android, "Android NDEF Push Protocol Specification", 2011.

[25] ISO/IEC, "ISO/IEC 14443 Contactless interface introduction.", 2006.

[26] ISO/IEC, "ISO/IEC 15693: Identification cards, Contactless integrated circuit cards, Vicinity cards", 2010.

[27] NFC Forum, "Logical Link Control Protocol, Tehnical Specification", 2009.

[28] NFC Forum, "NFC Data Exchange Format (NDEF)", Tehnical Specification", 2006.

[29] NFC Forum, "NFC Record Type Definition (RTD)", Tehnical Specification, 2006.

[30] NFC Forum, "Simple NDEF Exchange Protocol", Tehnical Specification", 2011.

[31] T. Ylonen, E. Lonvink, "RFC4254: The Secure Shell (SSH) Connection Protocol.", Technical report, 2006.

[32] (2013) Adrian Stabiszewski, Peer-to-peer communication using NFC in Android 4.4. Dostopno na:
`http://blog.opendatalab.de/hack/2013/11/25/android-p2p-nfc/`
(pridobljeno 3. 6. 2015)

[33] (2015) Android Studio Overview | Android Developers. Dostopno na:
http://developer.android.com/tools/studio/index.html
(pridobljeno 10. 1. 2015)

[34] (2015) Apple - Apple Pay. Dostopno na:
https://www.apple.com/apple-pay/
(pridobljeno 20. 1. 2015)

[35] (2014) Google Wallet. Dostopno na:
https://www.google.com/wallet/
(pridobljeno 7. 6. 2014)

[36] (2014) Installing Android SDK | Android Developers. Dostopno na:
https://developer.android.com/sdk/installing/index.html
(pridobljeno 22. 11. 2014)

[37] (2015) Java Software | Oracle. Dostopno na:
https://www.oracle.com/java/index.html
(pridobljeno 20. 4. 2015)

[38] (2014) The Legion of the Bouncy Castle. Dostopno na:
https://www.bouncycastle.org/
(pridobljeno 11. 12. 2014)

[39] (2014) MySQL. Dostopno na:
https://www.mysql.com/
(pridobljeno 5. 12. 2014)

[40] (2015) NetBeans. Dostopno na:
https://netbeans.org/
(pridobljeno 9. 1. 2015)

[41] (2015) Newtwork Basics. Dostopno na:
http://docs.oracle.com/javase/tutorial/networking/overview/networking.html
(pridobljeno 28. 1. 2015)

[42] (2015) NFC Forum. Dostopno na:
`http://nfc-forum.org/`
(pridobljeno 21. 11. 2014)

[43] (2015) Security Tips | Android Developers. Dostopno na:
`http://developer.android.com/training/articles/security-tips.html`
(pridobljeno 19. 2. 2015)

[44] (2015) Storage options | Android Developers. Dostopno na:
`http://developer.android.com/guide/topics/data/data-storage.html`
(pridobljeno 17. 4. 2015)

[45] (2014) Visa PayWave | Contactless. Dostopno na:
`http://www.visaeurope.com/receiving-payments/contactless`
(pridobljeno 5. 6. 2014)